What is Where by Looking: Weakly-Supervised Open-World Phrase-Grounding without Text Inputs

Tal Shaharabany **Yoad Tewel** **Lior Wolf**
Tel-Aviv University
{shaharabany,yoadtewel,wolf}@mail.tau.ac.il

Abstract

Given an input image, and nothing else, our method returns the bounding boxes of objects in the image and phrases that describe the objects. This is achieved within an open world paradigm, in which the objects in the input image may not have been encountered during the training of the localization mechanism. Moreover, training takes place in a weakly supervised setting, where no bounding boxes are provided. To achieve this, our method combines two pre-trained networks: the CLIP image-to-text matching score and the BLIP image captioning tool. Training takes place on COCO images and their captions and is based on CLIP. Then, during inference, BLIP is used to generate a hypothesis regarding various regions of the current image. Our work generalizes weakly supervised segmentation and phrase grounding and is shown empirically to outperform the state of the art in both domains. It also shows very convincing results in the novel task of weakly-supervised open-world purely visual phrase-grounding presented in our work. For example, on the datasets used for benchmarking phrase-grounding, our method results in a very modest degradation in comparison to methods that employ human captions as an additional input. Our code is available at `https://github.com/talshaharabany/what-is-where-by-looking` and a live demo can be found at `https://replicate.com/talshaharabany/what-is-where-by-looking`.

1 Introduction

> What does it mean, to see? The plain man's answer (and Aristotle's too) would be, to know what is where by looking. In other words, vision is the process of discovering from images what is present in the world, and where it is.
>
> Marr [50]

We address the task of detecting the objects that exist in a given image, without limiting ourselves to a predefined list of objects, of the type that detection algorithms employ. This property is sometimes referred to as open-world vision. Our method assigns a text phrase to each detected object. The similar task that employs an additional text input describing the image is called phrase-grounding. We describe our approach as "purely visual" to denote that there is no text input. Since the task is performed without the use of bounding boxes in any phase of training, the level of supervision employed corresponds to weakly supervised. As far as we can ascertain, the task of weakly-supervised open-world purely visual phrase-grounding is entirely novel. For short, we refer to this task, which is illustrated in Fig 1, as "what is where by looking" (WWbL).

In order to tackle this task we rely on two recent pretrained vision-language models: CLIP [60] and BLIP [41]. These models are trained on large corpora of matching images and captions and can produce a matching score between an input image and an input text. Trained on millions of such pairs, CLIP has been extensively used in zero-shot recognition [94], image generation [21, 25, 45], image

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

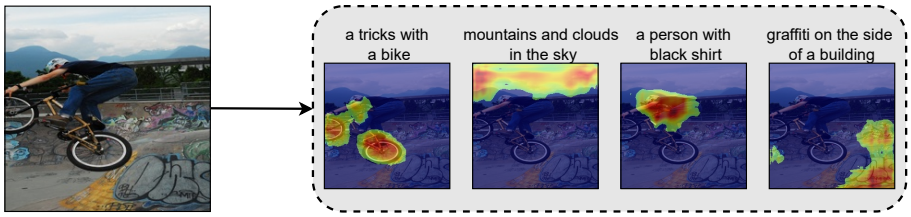

Figure 1: Given the image on the left, our method extracts multiple sentences that describe objects in the image, and generates a foreground mask that localizes the object

Table 1: Comparison to different lines of previous work on image localization.

Name	References	Open-world	Weakly supervised	Purely visual
Object detection	[61]	✗	✗	✓
Phrase-Grounding	[42, 85]	✓	✗	✗
Weakly supervised localization	[66]	✗	✓	✓
Weakly supervised Phrase-Grounding	[2]	✓	✓	✗
What is where by looking (WWbL)	This work	✓	✓	✓

editing and manipulation [57, 11, 24, 95], image and video captioning [53, 72, 73] and other tasks [52]. BLIP, a more recent addition, also incorporates into its training phase the ability to generate image captions. It also relies on a bootstrapped captioning model to generate matching sentences and on a bootstrapped filtering model to eliminate false matches from its training set.

While BLIP can produce a matching score, we choose to use CLIP for producing such a score and BLIP for generating plausible captions. This modular choice allows us to study the matching-based components (such as generalizing weakly supervised localization methods) separately from the WWbL. It also emphasizes the modular nature of our method. Given time and resources, we would be able to evaluate the method using only BLIP or replacing BLIP with CLIP-based image captioning methods, which are considerably slower than BLIP [73].

Our WWbL method comprises two phases. During training, an encoder-decoder g is applied to extract a foreground mask for an input image, given an image caption. This is the same level of supervision that is required to train CLIP or BLIP, albeit performed on a much smaller dataset. Since no segmentation mask (or bounding box) is used, this is considered weakly supervised.

Four loss terms are used. The first one requires that the CLIP matching score of the foreground region and the matching text be maximal. The second helps ensure that the background region is unrelated to the text. The third term considers the explainability map of CLIP, given the input image and the caption [13] as a guide for the foreground mask. Lastly, in order to encourage compact foreground masks, a regularization term is added.

A two-stage inference procedure employs the learned foreground segmentation network g and a pretrained-BLIP model. The first stage extracts object proposals using a Selective Search algorithm [74]. The second stage uses BLIP to generate captions for each proposal and find text clusters in CLIP's embedding space, in order to avoid repetitive descriptions. Network g is then used to find the foreground mask for each sentence, given the entire image as input and the sentence.

Our results show that network g, as a weakly supervised localization network, outperforms the state-of-the-art. The same model also outperforms the state-of-the-art in phrase grounding. When the entire method is applied, with the two stages described above, the results obtained for WWbL are on par with the results obtained using human captions for the task of phrase grounding. These results are considerably better than those obtained by the baselines.

Our main contributions are: (i) A new state-of-the-art solution for weakly-supervised localization tasks with text and image as input, and (ii) Tackling, for the first time, as far as we can ascertain, the fundamental WWbL task, which is an open-world detection task that does not rely on any localization information during training.

2 Related Work

Open-world computer vision is an emerging paradigm that stands in contrast to recognition systems that can only address the limited number of classes and types of examples that were observed during training. Specifically, open-set classification [7, 86, 38] requires the ability to understand when a new sample is not from the observed classes.

Since large visual-language models are trained on hundreds of millions of samples [60], many classes of objects are observed during training. It is impractical to enumerate all object classes. Moreover, these models can recognize phrases that were unseen during training, e.g., “a green dove”. Therefore, the use of the term "open-world" for such models is not based on learned vs. unseen classes, but rather on testing the method without limiting the type of objects observed during the inference.

Open-world localization methods generalize more restricted methods, which rely on a relatively short list of detectable objects. Similarly, weakly supervised methods, which employ labeling at the image level, generalize fully supervised methods that require bounding boxes during training. Finally, methods that do not receive as input guiding information regarding the content of the image (we call these “purely visual”) generalize methods that receive as input a list of objects to detect or text that describes the contents of the image.

Tab. 1 compares the task we tackle to other localization tasks. As can be seen, the task we tackle -“what is where by looking” (WWbL) - generalizes all other localization tasks. Unlike supervised object detection, it is open-set and weakly supervised. Weakly supervised localization methods are not open-set. Lastly, phrase grounding methods employ a textual input, while WWbL does not.

Object Detection One of the fundamental tasks of computer vision is object detection, where a closed set of object categories is localized based on training sets that contain costly bounding-box annotation [43, 20]. Since the advent of deep learning, many fully supervised algorithms have been proposed for this task [61, 26, 44]. In contrast to these methods, our approach employs image-level annotation (global text description) and an open dictionary to describe the detected object.

Phrase Grounding is a weakly supervised phrase localization task, in which text phrases are associated with local image annotations and local human-annotation signal [2, 89, 33]. Many methods extract text embedding from a pretrained language model [16, 63, 35], together with image representations, to obtain a common semantic space for image-text pairs. Our model is based on CLIP [60] as a text-image relation space to extract the localization of objects. Li et al. [42] collected 27M image-text data points, of which 3M have local human annotation, and used a semi-supervised training method for phrase-grounding, which employs CLIP. It cannot be compared directly with our work, since it uses additional supervision. Arbelle et al. [3] proposed a self-supervision method scheme for generating a localization map.

Weakly Supervised Localization (WSOL) Class Activation Map (CAM) explainability methods have been offered in recent years for solving WSOL tasks [92, 93, 59, 46]. Most of these algorithms train a classifier to distinguish between sub-categories of the main object (Birds, Cars, Dog etc), employing a localization loss term for the explainability map [80, 82, 51, 28, 55].

Similarly to us, Shaharabany et al. [66] employ an external localization network in addition to the classifier. During training, the classifier supplies gradients to the localization network. In contrast to our work: (1) they compare the classifier outputs with and without the foreground mask, while our algorithm compares the masked image with the text in the CLIP space, (2) there is no background loss term, and (3) explainability is not used as a signal.

Image Captioning Captioning is a fundamental vision-language task. Early methods applied RNNs [49, 37]. Attention was added to identify relevant salient objects [83, 64]. Subsequently, Transformers modeled interactions among all image elements with self-attention [75, 56, 22]. Recent works have shown significant improvements in robustness and generalization by using large-scale vision-language data sets [73, 53, 41, 34, 77]. Our method employs BLIP [41], which is a vision-language pre-training framework that is trained on images and their caption, employing a method to filter out noisy data. The framework also provides an image captioning model and shows good performances in diverse image domains and for a large variety of real-world objects.

Explainability Many methods generate a heatmap that indicates relevancy for CNNs, e.g., [65, 8, 47, 67]. Most relevant to our work is the use of GradCAM [65] relevancy maps as a cue for weakly

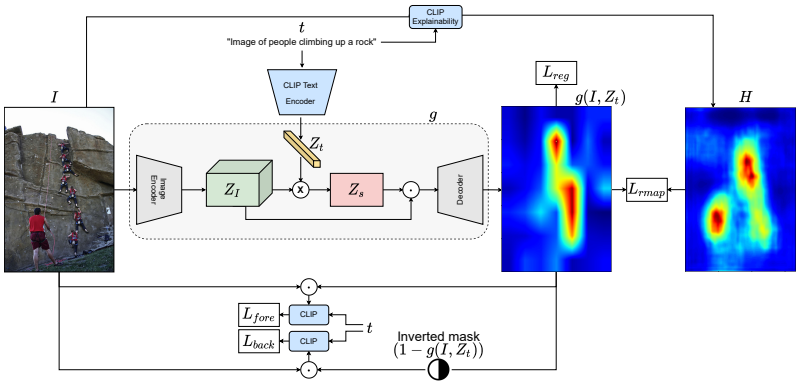

Figure 2: The architecture of our model and the optimization loss terms. The input image I is encoded with the VGG16 encoder part of g, and the test description t is encoded with CLIP's text encoder. The resulting text embedding multiplies the spatial dimension of the image embedding to find a similarity between regions in the image. The similarity map Z_s weights the image embedding. The results are passed to the decoder part of g to generate the output map. The loss term L_{rmap} compares the output map to the CLIP explainability map with L2 loss. L_{fore} forces the output map to focus on the text-relevant pixels by increasing the similarity between $I \odot g(I)$ and the text. L_{back} aims to guide the 1-$g(I)$ to focus on the background by decreasing the cosine similarity between $I \odot 1 - g(I)$ and the text. L_{reg} adds a sparsity constraint on $g(I)$

supervised segmentation methods [54, 79], including recent contributions that employ GradCAM heatmaps as pseudo labels for weakly supervised localization and segmentation algorithms[78, 28]. The literature on transformer explainability is sparse and is mostly limited to pure self-attention architectures [1, 12]. A recent work by Chefer et al. [12] fully addresses bi-modal transformer networks, and is the method employed to obtain relevancy maps. The solution is based on Layer-wise Relevance Propagation [5], with gradient integration for the self-attention and co-attention layers.

3 Method

In this section, we describe our method for generating a localization map based on image and text input signals without local annotation. We use our method on three different tasks (i) Weakly supervised object localization (WSOL) (ii) Weakly supervised phrase-grounding (WSG) (iii) What is where by looking (WWbL).

At the core of our method lies a segmentation network g, which is trained in a weakly supervised manner. The network g has two inputs: an input image $I \in R^{3\times W\times H}$, and an embedding of the input text Z^T. For the simpler task of WSOL, network g receives only the first input I as input signal. The loss terms used in both cases are the same, and, for the sake of generality we write the equations with g that has two inputs.

The training of g does not involve any ground truth localization information. Instead, it relies on two types of signals: (1) comparing the CLIP score of the extracted segment to that of the entire image, and (2) including the heatmap of the CLIP network, as obtained with the relevancy method of Chefer et al. [14], in the segmented region.

Four-loss terms are used, using the input image I, the text t, and the relevancy heat-map of CLIP given I and t, which we denote as H. The terms are (1) a foreground loss $L_{fore}(I, t)$ (2) a background loss $L_{back}(I, t)$, (3) a relevancy heatmap loss $L_{rmap}(I, H)$, and (4) a regularization loss $L_{reg}(I, g)$.

The foreground loss encourages the segmentation network g to output a map that increases the similarity between the text and the masked image $g(I, t) \odot I$, $\odot$ being the Hadamard product.

$$L_{fore}(I, t) = -CLIP(g(I, Z^T) \odot I, t), \tag{1}$$

where $CLIP(J, s)$ denotes the CLIP matching score between an image J and the text s.

The background loss helps ensure that the complementary part to that returned by $g(I, Z^T)$ is irrelevant, in the sense that its CLIP matching score with the text is low.

$$L_{back}(I, t) = CLIP((1 - g(I, Z^T)) \odot I, t), \tag{2}$$

The relevancy map H obtained by the method of Chefer et al. [14] on $CLIP(I, t)$ provides us with another important cue. While it is not entirely reliable in the sense that irrelevant regions are often marked and those regions often do not include the entire relevant part, we find that it often highlights some of the relevant parts. Since both H and $g(I, Z^T)$ have the same range of values ([0,1]), we employ the squared Euclidean norm of the difference.

$$L_{rmap}(I, H) = \|H - g(I, Z^T)\|^2 \tag{3}$$

The background loss may encourage $g(I, Z^T)$ to be as inclusive as possible, so that the complement is minimal. Additionally, the relevancy map loss L_{rmap} may encourage the obtained foreground mask to be overly inclusive. To encourage g to be more spatially limited, we add a regularization loss

$$L_{reg}(I, g)) = \|g(I, Z^T)\| \tag{4}$$

The combined loss is defined as $L = \lambda_1 * L_{fore}(I, t) + \lambda_2 * L_{back}(I, t) + \lambda_3 * L_{rmap}(I, H) + \lambda_4 * L_{reg}(I)$, where $\lambda_1, ..., \lambda_4$ are fixed weighting parameters for all datasets, which were determined after a limited hyperparameters search on the CUB [76] validation set to be 1, 1, 4, 1 respectively.

3.1 Inference Tasks

Weakly supervised object localization (WSOL) The goal of the WSOL task is to localize a single object in the image I without local annotation. For each fine-grained dataset, there is a single type of object (birds, cars, dogs, etc), and we do not guide the segmentation process with any type of conditioning. The text is used only in the training time for the loss functions, and is simply the name of the object ("birds", "cars", "dogs").

For converting the obtained continuous segmentation map $g(I)$, which is a sigmoid output between 0 to 1, to a binary mask, we follow the binarization used, e.g., by Qin et al. [59, 18, 17, 66], which is the standard in the benchmark. Namely, low-value pixels, below the threshold of 0.1, are zeroed out. Then, the method of Suzuki et al. [70] is used to extract contours. From the largest contour found, the algorithm selects the bounding box of the object.

Weakly supervised phrase-grounding (WSG) The aim of weakly supervised phrase-grounding is to generate a localization map, given an image I and a textual phrase t for a specific object in the image. To extract the objects from the output map of $M = g(I, Z^T)$, we first zero low-value pixels, using a threshold of 0.5. Next, we find contours with the method of Suzuki et al. [70]. For each contour, we extract the proper bounding box. The score for each box is calculated from the output map in the positions of the detected box. In the final stage, we apply non-maximal suppression over the boxes with 0.3 IoU, and filtering boxes with a low score (50% less) compared to the maximum score. See Appendix D for the visualization of these steps.

WWbL Inference The WWbL task is an open-world localization task, in which the only input is a given image. The task consists of localizing and describing all the elements composing the scene. To solve this task, we propose a two-stage algorithm based on a selective search [74] algorithm to extract object proposals to describe local regions in the image and incorporate the learned network g, the image captioning network BLIP, and the CLIP model.

Captioning models usually tend to describe the main element of the scene, neglecting other objects or the background. To address this issue, our proposed algorithm extracts local regions from the image with the Selective Search [74] procedure, which first over-segments the image into super-pixels, and then groups those in a bottom-up manner to propose candidate bounding boxes, see line 2 of Alg. 1. Each region is passed through BLIP to generate a local caption (line 3).

To avoid duplicate captions, the algorithm projects the captions to CLIP space (line 4) and applies the Community Detection (Cd) algorithm [9] to find caption clusters (line 5). The clustering algorithm uses cosine similarity in CLIP space between each caption such that a matrix of size $M \times M$ is obtained, where M is the number of captions. The algorithm employs two hyperparameters: (i) cosine similarity threshold for items in the same cluster and (ii) minimum size of a cluster such that, for each caption (row in the matrix), we find the number of captions above the threshold and check whether it is above the minimum size for a cluster. The final stage of Cd filters out overlapping clusters, giving priority to larger clusters. We set the cosine similarity threshold to be 0.85, and the minimum size to

Algorithm 1: WWbL inference method

Require: Input image I
Load trained networks g, BLIP and E_t
$\{P_i\}_{i=1}^{n} = SelectiveSearch(I)$
$\{T_i\}_{i=1}^{n} = BLIP(\{P_i\}_{i=1}^{n})$
$\{Z_i^T\}_{i=1}^{n} = E_t(\{T_i\}_{i=1}^{n})$
$\{T_i, Z_i^T\}_{i=1}^{m} = Cd(\{Z_i^T\}_{i=1}^{n}, 2, 0.85)$
$D \leftarrow \phi$
for $i = 1....m$ **do**
    $M_i = g(I, Z_i)$ ▷ Generate map
    $B_i \leftarrow BE(M_i)$ ▷ Extract bounding box
    $D = D \cup (B_i, T_i)$ ▷ Add object
return D

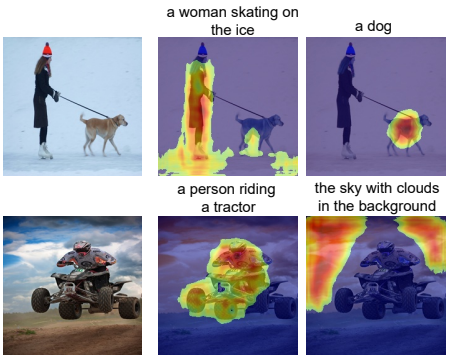

Figure 3: Sample WWbL results for Filckr30K

be 2. This algorithm returns a caption for each cluster and its embedding in CLIP space. For each cluster, g generates a localization map. Next, the bounding boxes are extracted like in Sec.3.1.

3.2 Architecture

Our network g is used in two types of tasks where (1) image captioning is used only for the loss function (WSOL) and g receives only the image (2) image captioning is used as input to g and in the loss function (WSG and WWbL). In this section, we describe each mode.

WSOL Architecture For WSOL, Network g is based on a U-Net [62] architecture, with skip connections between the encoder and the decoder for signals with the same spatial resolution. The decoder of g employs five upsampling layers, with each block containing two convolutional layers with batch normalization after the last convolutional layer before the activation function. The last activation function is Sigmoid, while the others are Relu.

Multi-Model Architecture For WSG and WWbL, Network g is based on an encoder-decoder architecture adapted to support text-based conditioning. This is illustrated in Fig. 2.

The encoding Z_t for the input text t is produced by the CLIP text encoder, where the size of the vector Z_t is 512. The image encoder of g is a VGG network [68], in which the receptive field is of size 16×16, i.e., the image is downscaled four times. The number of channels of the obtained map Z_I is also 512. The VGG network is initialized using an ImageNet pre-trained model.

We then consider the vector in $\mathbb{R}^{512}$ associated with each spatial location of the tensor Z_I. It is normalized to have a norm of one. The dot product with the vector Z_t (which in CLIP is also normalized) is computed. Performing this over all spatial locations, we obtain a map Z_s with values between -1 and 1. Z_s has a single channel and the same spatial dimensions as Z_I.

Z_s can be viewed as a text-conditioned importance map. All channels of Z_I are multiplied by Z_s and the result is passed to the decoder part of g. This decoder consists of three upsampling blocks, each with a sampling factor of two. Each block contains two convolutional layers with a kernel size equal to 3 and zero padding equal to one. Batch normalization is used after the last convolution layer, before the activation function. The first layer's activation function is a ReLU, while the last layer's activation function is a Sigmoid.

4 Experiments

We present our results for the three tasks: (i) weakly supervised object localization (WSOL) , (ii) weakly supervised phrase grounding (WSG), with training on either MSCOCO 2014 [43] or the Visual Genome (VG) dataset [40], and (iii) the new task we present (WWbL). For the first task, we employ three fine-grained localization datasets, and for the other two, we use the three datasets commonly used in WSG.

Datasets For the task of WSOL we evaluate our model on fine-grained datasets often used for this task. CUB-200-2011 [76] contains 200 birds species, with 11,788 images divided into 5994 training images and 5794 test images. Stanford Car [39] contains 196 categories of cars, with 8144 samples

Table 2: WSOL results for the CUB dataset.

Method	Backbone	Accuracy[%]
GAE [12]	CLIP	68.01
CAM [92]	VGG16	55.10
ACoL [90]	VGG16	62.96
ADL [19]	VGG16	75.41
DANet [84]	VGG16	67.70
MEIL [48]	VGG16	73.84
GCNet [46]	VGG16	81.10
POSL [87]	VGG16	89.11
SPA [55]	VGG16	77.29
SLT [28]	VGG16	87.60
FAM [51]	VGG16	89.26
ORNet [82]	VGG16	86.20
BAS [80]	VGG16	91.07
Ours	VGG16	**93.94**
CAM [92]	MobileNetV1	63.30
HaS [69]	MobileNetV1	67.31
FAM [51]	MobileNetV1	85.71
RCAM [6]	MobileNetV1	78.60
BAS [80]	MobileNetV1	92.35
Ours	MobileNetV1	**94.40**
CAM [92]	Resnet50	57.35
POSL [87]	Resnet50	90.00
WTL [4]	Resnet50	77.35
FAM [51]	Resnet50	85.73
SPOL [78]	Resnet50	96.46
BAS [80]	Resnet50	95.13
Ours	Resnet50	**96.54**
CAM [92]	InceptionV3	55.10
DANet [84]	InceptionV3	67.03
I2C [91]	InceptionV3	72.60
GCNet [46]	InceptionV3	75.30
SPA [55]	InceptionV3	72.14
SLT [28]	InceptionV3	86.50
FAM [51]	InceptionV3	87.25
BAS [80]	InceptionV3	92.24
Ours	InceptionV3	**94.30**
PSOL [88]	DenseNet161	93.01
Ours	DenseNet161	**93.71**

Table 3: Localization accuracy (percentage) WSOL on the cars and dogs datasets.

Method	Stanford-Cars	Stanford-Dogs
CLIP [12]	81.2	64.7
CAM [92]	65.2	66.0
HaS [92]	87.4	77.5
ADL [19]	82.8	73.5
RDAP [17]	92.9	77.7
FG [66]	96.2	79.2
Ours	**98.9**	**86.4**

Table 4: Backbones comparison for weakly supervised segmentation, as evaluated on CUB

Backbone	Params[#]	Accuracy[%]
Resnet18	11.1M	94.85
Resnet34	21.3M	95.31
Resnet50	23.5M	96.54
Resnet101	42.5M	96.64
Hardnet39DS	3.5M	93.77
Hardnet68DS	4.2M	93.85
Hardnet68	17.6M	94.99
Hardnet85	36.7M	95.11

Table 5: Ablation results for weakly supervised localization on CUB[76].

$R(g(I))$	L_{rmap}	L_{fore}	L_{back}	Accuracy[%]
-	-	-	√	48.36
-	-	√	-	47.58
-	-	√	√	42.11
√	-	-	√	69.01
√	-	√	-	84.72
√	-	√	√	90.02
-	√	-	-	87.02
-	√	-	√	89.12
-	√	√	-	90.09
-	√	√	√	91.08
√	√	-	-	84.24
√	√	-	√	92.67
√	√	√	-	94.73
√	√	√	√	96.54

in the training set and 8041 samples in the test set. Stanford dogs [36] consists of 20,580 images, with a split of 12,000 for training and 8580 for testing, where the data has 120 classes of dogs.

For the task of WSG and WWbL, our model trains on the split of MSCOCO and VG, and we evaluate it on the test splits of Flickr30k, ReferIt, and VG. MSCOCO 2014 [43], using the splits of Akbari et al. [2], consists of 82,783 training images and 40,504 validation images. Each image has five captions describing it. VG [40] contains 77,398 training, 5000 validation, and 5000 test images. Each image comes with a set of free-form text annotated bounding boxes.

Flickr30k [58] Entities, which is based on Flickr30k, contains 224K phrases describing objects in over 31K images, each described by 5 captions. For evaluation, we use the same 1k images from the test split of Akbari et al. [2]. ReferIt [27, 15] contains 20K images and 99,535 segmented images that also contain a description for the entire image. These image regions were collected in a two-player game with approximately 130K isolated entity descriptions. We use the split of Akbari et al. [2].

Implementation details In WSOL, our Algorithm receives an input image of size 224×224. During training, the image is first resized to 256×256 and then a random crop of size 224×224 is extracted. During the evaluation, the image is resized to 224×224.

Network g for WSOL is based on an Imagenet pretrained visual encoder. We compare the performance obtained using VGG16 [68], MobileNetV1 [30], Resnet50 [29], InceptionV3 [71] and

Table 6: WSG results: "pointing game" on VG, Flickr30K, and ReferIt.

Method	Backbone	VG trained			MS-COCO trained		
		VG	Flicker	ReferIt	VG	Flicker	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 [12]	CLIP	54.72	72.47	56.76	54.72	72.47	56.76
FCVC [23]	VGG	-	-	-	14.03	29.03	33.52
VGLS [81]	VGG	-	-	-	24.40	-	-
TD [89]	Inception-2	19.31	42.40	31.97	-	-	-
SSS [33]	VGG	30.03	49.10	39.98	-	-	-
MG [2]	BiLSTM+VGG	50.18	57.91	62.76	46.99	53.29	47.89
MG [2]	ELMo+VGG	48.76	60.08	60.01	47.94	61.66	47.52
GbS [3]	VGG	53.40	70.48	59.44	52.00	72.60	56.10
ours	CLIP+VGG	**62.31**	**75.63**	**65.95**	**59.09**	**75.43**	**61.03**

Table 7: WWbL and WSG results: "pointing game" and bounding box accuracy.

Task	Model	Training	Test Point Accuracy			Test Bbox Accuracy		
			VG	Flickr	ReferIt	VG	Flickr	ReferIt
WWbL	MG[2]	COCO	32.91	50.12	36.34	11.48	23.75	13.31
	MG[2]	VG	32.15	49.48	38.06	12.23	24.79	16.43
	GAE [12]	-	38.15	56.25	41.64	9.69	17.14	12.31
	ours	COCO	**44.20**	**61.38**	43.77	17.76	**32.44**	**21.76**
	ours	VG	43.91	58.59	**44.89**	**17.77**	31.46	18.89
WSG	MG [2]	COCO	47.94	61.66	47.52	15.77	27.06	15.15
	MG [2]	VG	48.76	60.08	60.01	14.45	27.78	18.85
	GAE [12]	-	54.72	72.47	56.76	16.70	25.56	19.10
	ours	COCO	59.09	75.43	61.03	27.22	35.75	30.08
	ours	VG	**62.31**	**75.63**	**65.95**	**27.26**	**36.35**	**32.25**

DenseNet161 [31] is applied for CUB dataset to compare performances with the state-of-the-art baselines, while for the other datasets we used Resnet50, since that is what the literature baselines have published. A SGD optimizer with a batch size of 48 and an initial learning rate of 0.0003 for 100 epochs is used. The optimizer momentum of 0.9 and weight decay of 0.0001 are also used. During the training, a random horizontal flip with 0.5 probability is applied. All models are trained on a single GeForce RTX 2080Ti Nvidia GPU. In WSG training (the same network is used for inference in both WSG and WWbL), following Akbari et al. [2], network g receives an input image of size 299×299 and generates a localization map of the same size. During training, the image is also resized to 224×224, which is the input size of CLIP. The relevancy map H is generated at this resolution and is resized to 299×299. Network g for WSG and WWbL is VGG16 [68], to ensure a fair comparison with the baselines. An SGD optimizer (batch size of 32 and an initial learning rate of 0.0003) is used for 100 epochs. The optimizer momentum of 0.9 and weight decay of 0.0001 are also used. During training, a random horizontal flip is applied, with 0.5 probability. All models are trained on a double 2080Ti Nvidia GPU, while all experiments of Arbelle et al. [3] were conducted on four V100 GPUs.

Results For the WSOL task, the main accuracy metric measures whether the intersection over union (IoU) of the ground-truth bounding box and the detector's output are above 0.5. As can be seen in Tab. 2, our method obtains state-of-the-art performance for the CUB [76] dataset, for all five different backbones. The method also outperforms GAE [12] with CLIP as a backbone. Our method improves the GAE map, which it uses during training, by more than 25% in the accuracy metric. The results listed in Tab. 3 show that our method also achieves state-of-the-art performance for the fine-grained localization datasets. Following previous work, a Resnet-50 is used for these methods, except for GAE, which is applied with a CLIP backbone. Appendix E presents sample results.

For the WSG task, the algorithm is evaluated with respect to the accuracy of the pointing game [89], which is calculated from the output map by finding the maximum-value location for the given query and checking whether this point is located in the region of the object. Another metric we report (B-Box accuracy) compares the extracted bounding-boxes with the bounding-box annotations in the same manner as the WSOL task above. For a fair comparison, we use the same training/validation/test splits as Akbari et al. [2]. For ReferIt, Visual Genome and Flickr30K, we treat each query as a single sentence. Tab. 6 summarises the results for the Flickr30k, ReferIt and Visual Genome datasets for the WSG task. Evidently, our method is superior to all baselines, whether training takes place over VG or MS-COCO. Tab. 7 presents bounding box accuracy results for the WSG task. Here, too, our method outperforms the baseline methods. In appendix B we present sample output maps.

For the WWbL task, we use the same metrics (pointing game and bounding-box accuracy). For each ground-truth pair of bounding box and caption, we select the closest caption in CLIP space from the output predictions and compare the output map to the bounding box using the pointing accuracy metric. In addition, bounding boxes are extracted for the heat-map, as described at Sec. 3.1, and compared to the selected ground-truth bounding boxes with the same accuracy metric as above.

Tab.7 presents the results for the WWbL task. In addition to our method (Alg. 1) that utilizes network g to generate a heat-map, given an image and a text (line 8), we also test two alternative heatmap generation methods. The first one is the MG method of Akbari et al. [2] for the WSG task, and the second is the GAE explainability method [12]. As can be seen in the results listed in Tab. 7, our proposed method obtains the best results among the three. For comparison, we also provide WSG results using Alg. 1 and the ground truth text for the same three heatmap methods. Interestingly,

Figure 4: WWbL on a few samples of web images.

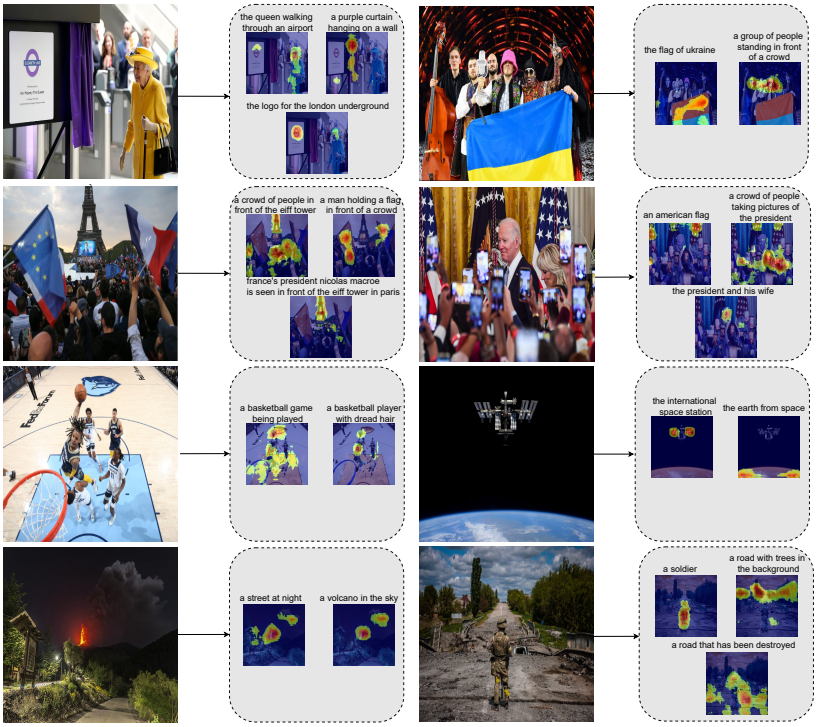

Figure 5: WWbL on images from Week in pictures by BBC (April-May 2022).

the g that is trained on the VG dataset shows better performance for WSG, while in WWbL the results are mixed. Fig 3, Fig. 4 show results for our method from Flickr30K and web images samples respectively. Fig. 5 shows results for our method from BBC - Week in pictures from the recent weeks (April-May 2022).

Ablation Study In this section, we compare the accuracy of g with different modes of the loss function and present the experiment used to determine the values of $\lambda_{1,2,3,4}$. In addition, we also

compare the performance of two backbone families, Resnet [29] and HarDnet [10]. All experiments were conducted in the WSOL settings for the CUB[76] dataset. Tab. 5 presents the sensitivity of our network g to each loss term, as evaluated on the test set, using the Resnet50 backbone. The first part of the table compares the performance without L_rmap, using different combinations of the other loss terms. As can be seen, the best performance is achieved when we use the three terms $L_{fore}, L_{back}, L_{reg}$ together. Each term improves the performance: L_{fore} improves it by 21%, L_{back} by 5% and L_{reg} by 48%. The second part compares the performance with L_{rmap} which improved by more than 6%. As can be seen, each of the other terms improves the performance: L_{fore} improves it by 4%, L_{back} by 2%, and L_{reg} by 5.5%. Also, training only with L_{rmap} decreases the performance by more than 9% compared to performance using all loss terms. The backbone ablation was conducted with two backbone families and is summarised in Tab. 4. For both families, performance improves as the number of parameters is increased.

Appendix C, presents the sensitivity of the network g to the λ_3 coefficient, which is the weight of L_{rmap} and the only coefficient that is not set to 1 in our experiments. The method seems to be largely insensitive to this parameter and it is stable for a wide range of values. The ability of g to generalize well under different pre-trained CLIP models is examined in App. G. Evidently, the variance in the performance of g is much lower than the variance in the zero-shot classification performance of the CLIP models used to train g.

5 Discussion and limitations

When biological systems see, the input is seldom limited to single images. Although our WWbL method was developed in the context of stills, it can be naturally extended to perform online localization of objects in video. For this, the detected regions and their matching concepts need to be tracked and must evolve over time, with the ability to correct previous hypotheses. We leave this effort for future research, since benchmarking this ability with the current datasets is challenging.

The common computer vision terminology makes the distinction between object localization and object detection. The former aims to locate the main object of a certain type in the input image, while the latter aims to find all the relevant objects and their individual boundaries. Our WWbL method aims to extract every existing object and can, therefore, detect all image objects.

The method employs the selective search method [74]. This can be avoided by identifying one region at a time and iterating the process. Such a method is presented in Appendix A. It uses g to recover the object map at each stage. The results reported in this Appendix are somewhat lower than the method based on a selective search. Since the image datasets and pre-trained models our method employs may be biased towards western media concepts, it is culturally biased. For similar reasons, it may display discriminatory behavior.

Another limitation is that our weakly supervised learning scheme does not distinguish between multiple instances of the same object. While Algorithm 1 can be improved to somewhat mitigate this, by separating multiple objects that have the same caption, building such a solution robustly may be challenging without additional supervision. App.F present visualization for this scenario.

6 Conclusions

Through the power of large pre-trained transformers and by integrating explainability cues, we build an effective weakly supervised phrase-grounding network g. By combining an off-the-shelf image captioning engine, we are able to identify and localize the objects within an input image. The task we solve is an open-world one and the setting generalizes multiple existing tasks. It is convincingly demonstrated that our weakly supervised solution is on par with the fully supervised alternatives that exist for the phrase-grounding task.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974).

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] 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.

[3] 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.

[4] Sadbhavana Babar and Sukhendu Das. Where to look?: Mining complementary image regions for weakly supervised object localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1010–1019, 2021.

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[6] Wonho Bae, Junhyug Noh, and Gunhee Kim. Rethinking class activation mapping for weakly supervised object localization. ECCV, 2020.

[7] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[8] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[9] 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.

[10] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. Hardnet: A low memory traffic network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3552–3561, 2019.

[11] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-based clip-guided essence transfer, 2021.

[12] 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.

[13] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] 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.

[15] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 824–832, 2017.

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

[17] 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.

[18] 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.

[19] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019.

[20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[21] Katherine Crowson. Vqgan-clip, 2021.

[22] 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. *ICLR*, 2021.

[23] 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.

[24] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators.

[25] Federico A Galatolo, Mario GCA Cimino, and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. *arXiv preprint arXiv:2102.01645*, 2021.

[26] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[27] 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.

[28] Guangyu Guo, Junwei Han, Fang Wan, and Dingwen Zhang. Strengthen learning tolerance for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7403–7412, 2021.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[32] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021.

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

[34] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters? low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*, 2021.

[35] 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.

[36] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2. Citeseer, 2011.

[37] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. *arXiv preprint arXiv:1411.7399*, 2014.

[38] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2021.

[39] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[40] 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.

[41] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[42] 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.

[43] 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.

[44] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, volume 9905 of *LNCS*, pages 21–37, 2016.

[45] Xingchao Liu, Chengyue Gong, Lemeng Wu, Shujian Zhang, Hao Su, and Qiang Liu. Fusedream: Training-free text-to-image generation with improved clip+gan space optimization. *ArXiv*, abs/2112.01573, 2021.

[46] Weizeng Lu, Xi Jia, Weicheng Xie, Linlin Shen, Yicong Zhou, and Jinming Duan. Geometry constrained weakly supervised object localization. *ECCV*, 2020.

[47] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[48] Jinjie Mai, Meng Yang, and Wenfeng Luo. Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8766–8775, 2020.

[49] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *CoRR*, abs/1412.6632, 2014.

[50] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[51] Meng Meng, Tianzhu Zhang, Qi Tian, Yongdong Zhang, and Feng Wu. Foreground activation maps for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3395, 2021.

[52] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021.

[53] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[54] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 685–694, 2015.

[55] Xingjia Pan, Yingguo Gao, Zhiwen Lin, Fan Tang, Weiming Dong, Haolei Yuan, Feiyue Huang, and Changsheng Xu. Unveiling the potential of structure preserving for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11642–11651, 2021.

[56] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *CVPR*, 2020.

[57] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[58] 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.

[59] 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.

[60] 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.

[61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[63] 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.

[64] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. High-order attention models for visual question answering. *NIPS*, 2017.

[65] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[66] Tal Shaharabany and Lior Wolf. Learning a weight map for weakly-supervised localization. *arXiv preprint arXiv:2111.14131*, 2021.

[67] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

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

[69] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, pages 3544–3553. IEEE, 2017.

[70] 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.

[71] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[72] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *MM*, 2021.

[73] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[74] 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.

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[76] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[77] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[78] Jun Wei, Qin Wang, Zhen Li, Sheng Wang, S Kevin Zhou, and Shuguang Cui. Shallow feature matters for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5993–6001, 2021.

[79] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[80] Pingyu Wu, Wei Zhai, and Yang Cao. Background activation suppression for weakly supervised object localization. *arXiv preprint arXiv:2112.00580*, 2021.

[81] 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.

[82] Jinheng Xie, Cheng Luo, Xiangping Zhu, Ziqi Jin, Weizeng Lu, and Linlin Shen. Online refinement of low-level feature based activation map for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 132–141, 2021.

[83] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[84] Haolan Xue, Chang Liu, Fang Wan, Jianbin Jiao, Xiangyang Ji, and Qixiang Ye. Danet: Divergent activation for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6589–6598, 2019.

[85] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021.

[86] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019.

[87] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. Rethinking the route towards weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13460–13469, 2020.

[88] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. Rethinking the route towards weakly supervised object localization. In *CVPR*, pages 13460–13469, 2020.

[89] 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.

[90] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, pages 1325–1334, 2018.

[91] Xiaolin Zhang, Yunchao Wei, and Yi Yang. Inter-image communication for weakly supervised localization. *arXiv preprint arXiv:2008.05096*, 2020.

[92] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[93] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[94] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.

[95] Peihao Zhu, Rameen Abdal, John C. Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. *ArXiv*, abs/2110.08398, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Will be included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A WWbl algorithm without selective search

In this section, we present an algorithm for wwbl without the selective search method for describing the objects in the image.

The algorithm runs N iterations with a break option when no object was found. In line 2 of the listed algorithm, we obtain the caption of the input image $T_0 = BLIP(I)$.

Lines 4-16 of Alg. 2 describe the iterative algorithm used to extract objects from a single image. The first operation is to encode the text in the current iteration with the CLIP text encoder. The next step is to generate a map with g. From the output map M_i we extract the largest bounding box (B_i), using the algorithm described in section 3.1. For the extracted bounding box, the algorithm crops a relative patch from the image (P_i), and updates the image, such that the patch pixels are zero. The new image is used for generating the new caption $T_i^P = BLIP(P_i)$. Next, the algorithm calculates the similarity between $E_t(T_0)$ and $E_t(T_i^P)$. Only if the similarity is above $\tau = 0.6$ does the algorithm add this bounding box to the predicted output and continues with the iterations. Otherwise, it stops the routine and returns the predicted output.

Tab. 8 compares the performances of algorithm 1 and 2. Evidently, while Alg. 2 addresses WWbL to some degree, Alg. 1 is considerably better.

Algorithm 2 WWbL Inference

Require: Input image I
1: Load networks g, CLIP, BLIP
2: $T_0 = BLIP(I)$
3: $D \leftarrow \phi$ ▷ Empty output dictionary
4: **for** $i = 0....n-1$ **do**
5: $Z_i = E_t(T_i)$ ▷ CLIP text encoder
6: $M_i = g(I, Z_i)$ ▷ Generate map
7: $B_i \leftarrow C(M_i)$ ▷ Extract bounding box
8: $P_i \leftarrow I[B_i]$ ▷ Crop patch
9: $T_i^P = BLIP(P_i)$ ▷ Caption patch
10: $S = CLIP(T_i^P, T_0)$
11: **if** $S \geq 0.6$ **then**
12: $D = D \cup (B_i, T_i^P)$ ▷ Add object
13: $I[B_i] = 0$ ▷ Delete the object
14: $T_{i+1} = BLIP(I)$ ▷ New caption
15: **else**
16: break
17: return D

Table 8: WWbL results: "pointing game" accuracy on Visual Genome (VG), Flickr30K, and ReferIt

Method	Backbone	Training	Test Accuracy		
			VG	Flickr30K	ReferIt
Algorithm 2	CLIP+VGG	VG	35.02	42.57	37.56
Algorithm 1	CLIP+VGG	VG	**43.91**	**58.59**	**44.89**
Algorithm 2	CLIP+VGG	COCO	32.80	41.86	37.08
Algorithm 1	CLIP+VGG	COCO	**44.20**	**61.38**	**43.77**

B Visualization of baselines - task WSG

We present visualization of our network g for different scenarios, together with the GAE method, to emphasize the refinement of our network, as can be seen at Fig. 6. The proposed loss terms for training g encourage the output mask to capture the whole shape of the object instead of specific regions of GAE which tend to focus more on discriminate regions.

Figure 6: Sample phrase-grounding results for our method compared with GAE of CLIP [12] where (a) the input image (b) our output map (c) GAE of CLIP

C λ_3 sensitivity

The proposed loss terms have four coefficients that balance the final loss function. To avoid an excessive search for hyperparameters, we set λ_1, λ_2 and λ_4 to one and only set λ_3 at a different value.

In Tab. 9 we study the effect of λ_3 on the performance of the learned g for the WSOL task on the CUB benchmark [76] and for WSG on Filckr30K. Evidently, the value of $\lambda_3 = 4$ is optimal for both experiments (this is the value used throughout our work). However, performance is relatively stable in a wide range of values. As noted in the ablation experiments of the paper, a value of zero (canceling this term), leads to a drop of 6% for WSOL on CUB. The current results indicate an even larger drop for Phrase grounding.

Table 9: The effect of varying λ_3 on the accuracy of weakly supervised localization, as evaluated on CUB, and on that of phrase grounding, evaluated on Flickr30K.

λ_3	0	0.01	0.1	1	2	4	8	16
CUB localization accuracy[%]	90.0	88.7	90.6	92.6	94.1	96.5	93.8	93.23
Phrase grounding accuracy [%]	41.9	41.3	44.5	61.8	73.1	75.6	75.4	75.1

D Visualization of bounding box extraction from localization map

In this section, we present visualization for bounding box extraction, which is used for WWbl, and also for WSG. We first set each pixel in the localization map that is below 50% under the maximum value in the map to zero. Next, we apply the algorithm of Suzuki et al. [70] to find the contours of all objects in the map. We consider the bounding box of each contour. We then calculate the sum of all values of the map g inside the bounding box, which we refer to as "energy". The final stage is to filter out bounding boxes via non-maximal suppression using the energy values, for all overlapping bounding boxes with an IOU larger than 0.3. The bounding boxes are then filtered further, keeping only the ones with energy values of at least 50% of that of the bounding box with the maximal energy.

Fig. 7 presents a visualization of the method where (a) the input image (b) the localization map (c) localization map after thresholding with bounding box proposals (d) localization map after thresholding with the final bounding boxes (e) the input image with the final bounding boxes. As can be seen, the first step of zeroing low-value pixels allows the method to focus on the relevant regions in the image based on the text. The importance of filtering non-relevant bounding boxes based on NMs and low energy also rises from the next examples.

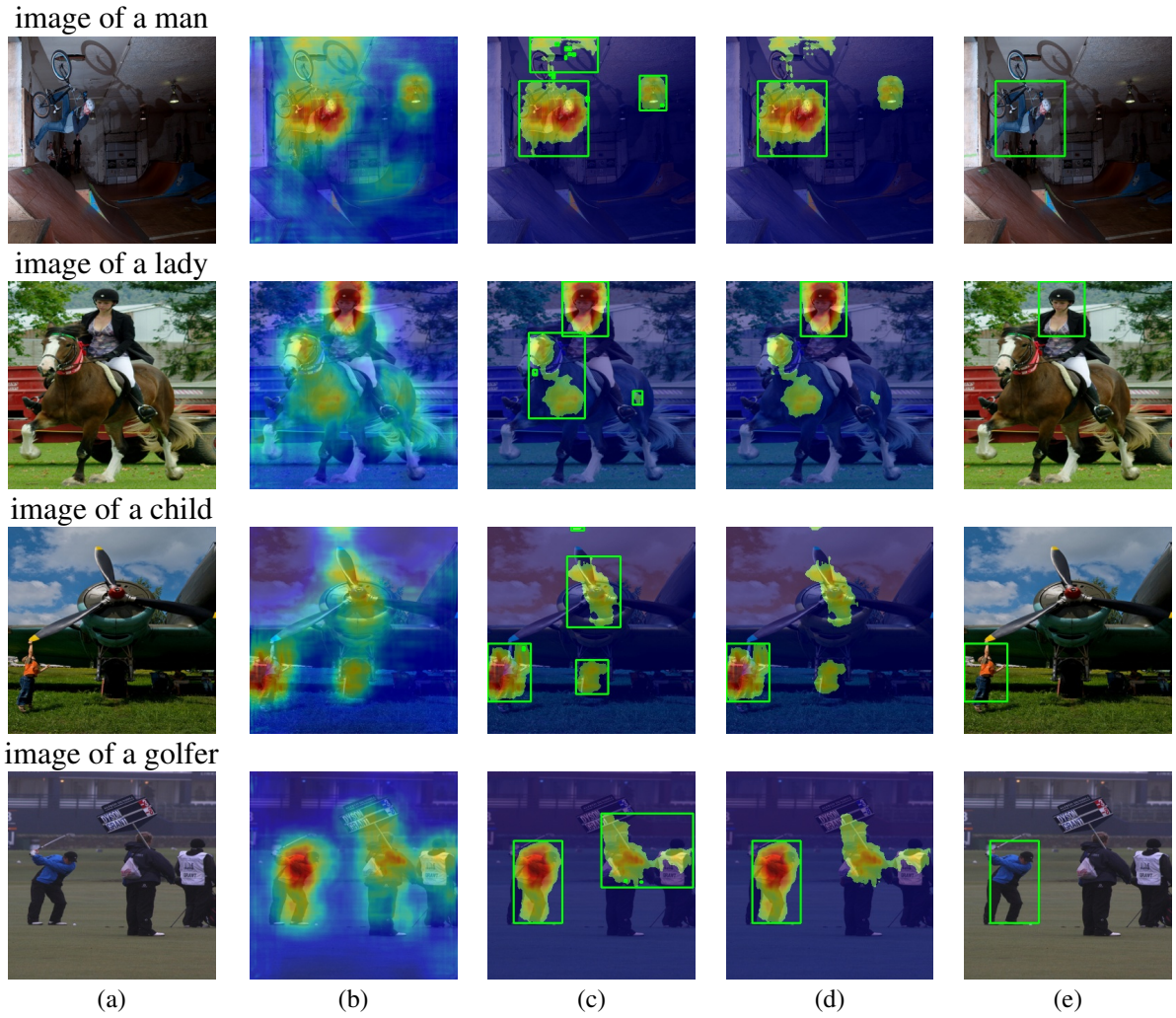

Figure 7: Visualization of bounding box extraction from localization map where (a) the input image (b) localization map (c) localization map after thresholding with bounding box proposals (d) localization map after thresholding with the final bounding boxes (e) the input image with the final bounding boxes

E WSOL visualization

In this section, we present a visualization of WSOL task for fine-grained datasets where the first row is the Stanford-cars dataset, the second row is the CUB dataset, and the last row is Stanford-dogs. Fig. 8 present our results together with GAE[12] which emphasizes our improvements for the relevant map of the GAE (this is used, during training only, as one of our losses) for the localization task.

F Multiple instances of the same object

In Fig. 9, we present our method's results for images with multiply instances of the same object, such as apples and dogs. On the left side, we present the results of the WWbL method. On the right side, we present results for grounding specific sentences that are provided to study the output of network g.

As can be seen, WWbL does not typically select sentences that distinguish between the various objects. However, network g can separate between different objects of the same class given specific captions. When there are multiple objects of the same type, e.g., multiple green apples, g marks all of them. We note that g's heatmap does peak at specific parts of the objects, which may facilitate instance separation (this direction is out of scope for the current study).

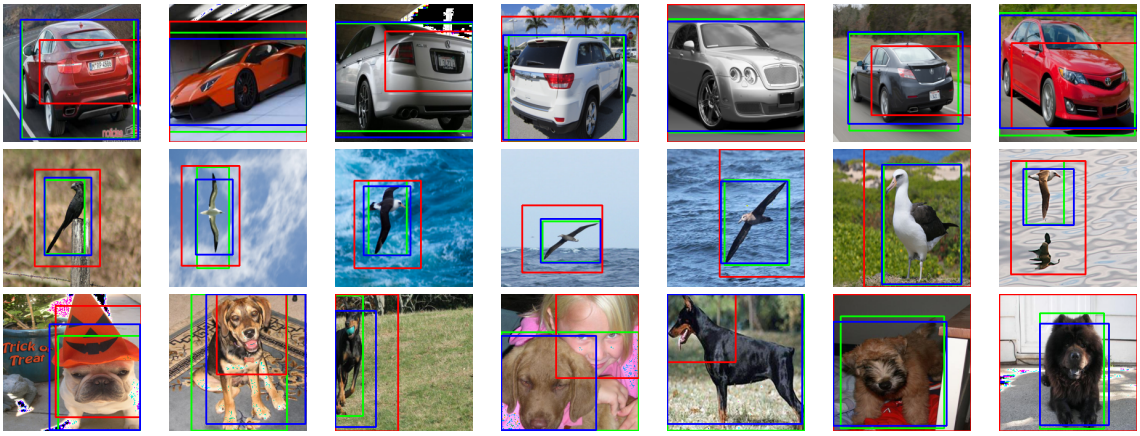

Figure 8: Sample localization results, where the first row is the visualization of the Stanford-cars dataset, the second row is the CUB dataset, and the last row is Stanford-dogs. The green bounding box is the ground-truth, the red is the bounding box of the CLIP explainability map, obtained wtih GAE, and the blue bounding box is the output of our algorithm.

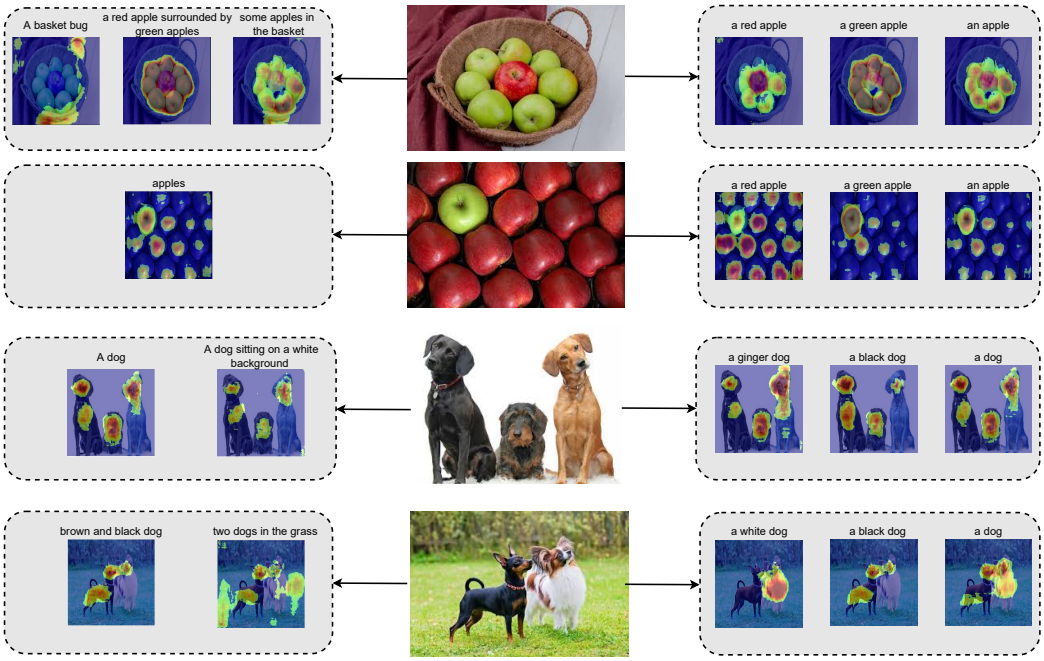

Figure 9: The heatmap obtained by network g for WWbL predictions (left side) and specific captions (right side).

G Employing different pre-trained CLIP models

In this section, we examine the ability of g to perform weakly supervised localization on CUB and Filckr30K for WSG task, when employing CLIP that is pre-trained on different datasets.

The models were taken from the OpenCLIP open-source repository [32], which provides both the pre-trained CLIP models and the obtained imagenet1K zero-shot accuracy.

As can be seen in Tab. 10, while the zero-shot classification varies considerably, the performance of g on the localization task varies much less. Also, there does not seem to be a strong correlation between the amount of training data used for training CLIP and the localization performance of g.

A similar experiment was conducted for the phrase grounding task, with similar results. As Tab. 10 indicates, the ResNet50 model slightly outperforms the results obtained by the ViT32 model, when both are trained on the openai dataset. Other datasets, both bigger and smaller, do not lead to better performance.

We note that the gap observed for both WSOL and Phrase Grounding is much narrower than the gap that we see in top-1 accuracy for zero-shot classification. This indicates that our method is relatively stable concerning the CLIP model used during training.

Table 10: Results obtained by network g in the task of (1) weakly supervised localization on the CUB benchmark, (2) phrase grounding on the Flicker benchmark, for different CLIP models. Also reported is the top-1 zero shot accuracy on ImageNet.

Clip version			Performance metrics		
Clip backbone	CLIP training dataset	Num. samples	CUB-Acc[%]	Flicker-Acc[%]	top1 ImageNet Acc[%]
RN50	openai	**400M**	89.3	**76.0**	**59.8**
RN50	yfcc15m	15M	88.1	72.9	32.4
RN50	cc12m	12M	**94.1**	73.3	35.9
RN101	openai	**400M**	**96.5**	**75.1**	**62.3**
RN101	yfcc15m	15M	87.2	69.5	34.0
ViT32	openai	400M	**96.5**	**75.6**	63.4
ViT32	laion2be16	**2B**	92.5	73.4	**65.6**
ViT32	laion400me32	400M	94.2	75.3	60.2