Taming Self-Training for Open-Vocabulary Object Detection

Shiyu Zhao[1,*] Samuel Schulter[2] Long Zhao[3] Zhixing Zhang[1]
Vijay Kumar B G[2] Yumin Suh[2] Manmohan Chandraker[2,4] Dimitris N. Metaxas[1]

[1] Rutgers University [2] NEC Laboratories America [3] Google Research [4] UC San Diego

Abstract

Recent studies have shown promising performance in open-vocabulary object detection (OVD) by utilizing pseudo labels (PLs) from pretrained vision and language models (VLMs). However, teacher-student self-training, a powerful and widely used paradigm to leverage PLs, is rarely explored for OVD. This work identifies two challenges of using self-training in OVD: noisy PLs from VLMs *and* frequent distribution changes of PLs*. To address these challenges, we propose SAS-Det that tames self-training for OVD from two key perspectives. First, we present a split-and-fusion (SAF) head that splits a standard detection into an open-branch and a closed-branch. This design can reduce noisy supervision from pseudo boxes. Moreover, the two branches learn complementary knowledge from different training data, significantly enhancing performance when fused together. Second, in our view, unlike in closed-set tasks, the PL distributions in OVD are solely determined by the teacher model. We introduce a periodic update strategy to decrease the number of updates to the teacher, thereby decreasing the frequency of changes in PL distributions, which stabilizes the training process. Extensive experiments demonstrate SAS-Det is both efficient and effective. SAS-Det outperforms recent models of the same scale by a clear margin and achieves 37.4 AP_{50} and 29.1 AP_r on novel categories of the COCO and LVIS benchmarks, respectively. Code is available at* `https://github.com/xiaofeng94/SAS-Det`.

1. Introduction

Traditional closed-set object detectors [4, 13, 32] are restricted to detecting objects with a limited number of categories. Increasing the size of detection vocabularies usually requires heavy human labor to collect annotated data. With the recent advent of strong vision and language models (VLMs) [20, 30], open-vocabulary object detection (OVD) [11] provides an alternative direction to approach this challenge. Typically, OVD detectors are trained with annotations of base categories and expected to generalize to novel categories with the power of pretrained VLMs.

One promising thread of recent studies for OVD [8, 9, 42, 49] leverages VLMs to obtain pseudo labels (PLs) beyond base categories. But they rarely explore self-training, a powerful and widely used schema for utilizing PLs in closed-set tasks [37, 38, 44, 45]. We investigate this and find the vanilla self-training approach does not improve OVD performance due to the following challenges.

First, the typical self-training in closed-set tasks sets a confidence threshold to remove noisy PLs based on the fact that the quality of PLs is positively correlated to their confidences. However, VLMs employed in OVD are pretrained for image-level alignment with texts instead of instance-level object detection that requires the localization ability. Thus, the confidence score from pretrained VLMs is usually not a good indicator for the quality of box locations (i.e., pseudo boxes) provided by PLs. For example, prior studies [11, 49] show that CLIP [30] tends to output imperfect object boxes as predictions with high confidence. Recent methods for OVD [9, 49] just apply thresholding to VLMs' confidence scores and ignore the poor quality of pseudo boxes, which provides noisy supervision to the model. This issue becomes even worse when self-training is applied directly, since the noise accumulates which degrades the performance on novel categories. Moreover, these methods handle noisy PLs in the same way as ground truth of base categories during training, which further decreases the performance on base categories [9, 23].

Second, self-training for closed-set object detection [19, 38, 45] usually follows an online teacher-student manner. In each training iteration, the teacher generates PLs, and the student is trained with a mixture of ground truth and PLs. Then, the teacher is updated by the student with exponential moving average (EMA). However, we find such EMA updates degrade OVD models (see Table 4). Our hypothesis is that, unlike closed-set tasks, OVD provides no ground truth for target categories, and thus, the supervision for target cat-

* Part of this work was done during an internship at NEC Laboratories America. Correspondence to: Shiyu Zhao sz553@rutgers.edu

egories is fully decided by the distribution of PLs predicted by the teacher. Hence, the EMA updates change the distribution of PLs in each iteration, unstabilizing the training.

In this paper, we propose *Self-training And Split-and-fusion head* for open-vocabulary Detection (SAS-Det) to tame self-training for OVD. First, we present a split-and-fusion (SAF) head to handle the noise of PLs. The SAF head splits the standard detection head into two branches: the closed-branch and the open-branch, which are fused at inference. The closed-branch, akin to the standard detection head, comprises a classification module and a box refinement module. It is supervised solely by ground truth from base categories, mitigating the impact of noisy PLs on the performance of base categories. The open-branch is a classification module supervised by class labels of both ground truth and PLs. It acquires complementary knowledge to the closed-branch and can significantly boost the performance when fused with the closed-branch. Moreover, this design circumvents noisy locations of pseudo boxes, reducing the accumulation of noise during self-training.

Second, instead of adopting the vanilla EMA update, we reduce the number of the updates and periodically update the teacher by the student. The quality of our PLs improves along with the periodic updates, and our final PLs are better than those of prior PL-based methods [9, 49] that introduce external handcrafted steps. Fig. 1 shows the key differences.

The proposed SAS-Det outperforms recent OVD models of the same scale by a clear margin on two popular benchmarks, i.e., COCO and LVIS. Without extra handcrafted steps, our pseudo labeling is more efficient than prior methods, i.e., nearly 4 times faster than PB-OVD [9] and 3 times faster than VL-PLM [49]. Extensive ablation studies demonstrate the effectiveness of the proposed components. On COCO, the thresholding of vanilla self-training decreases the performance of novel categories by 3.6 AP, and the EMA update decreases the performance by 6.9 AP. Instead, SAS-Det eliminates the degradation with two separate detection heads and the periodic update. The fusion of the SAF head boosts the performance by 6.0 AP.

The contributions of this work are summarized as follows. (1) We show two challenges of applying self-training to OVD and propose two simple but effective solutions, i.e., using different detection heads to mitigate the noise in PLs, and using periodic updates to reduce frequency of changes in PLs' distributions. (2) The proposed SAF head for OVD handles the noisy boxes of PLs and enables fusion to improve the performance. (3) We present the leading performance on COCO and LVIS under widely used OVD settings and provide detailed analysis of the proposed SAS-Det.

2. Related Work

Vision-language models (VLMs). VLMs are trained to learn the alignment between images and text in a common embedding space. CLIP [30] and ALIGN [14] use contrastive losses to learn such alignment on large-scale noisy image-text pairs from the Internet. ALBEF [20] introduces multi-modal fusion and additional self-supervised objectives. SIMLA [16] employs a single stream architecture to achieve multi-level image-text alignment. FDT [6] learns shared discrete tokens as the embedding space. These VLMs achieve impressive zero-shot performance on image classification. But due to the gap between the pretraining and detection tasks, VLMs have limited abilities in object detection. In this work, we attempt to close the gap via PLs. There are studies [15, 21, 34] focusing on aligning any text phrases with objects. But they require visual grounding data that are more expensive than detection annotations. Our work scales up the vocabulary size for object detection without requiring such costly data.

Open-vocabulary object detection (OVD). Zero-shot object detection methods [3, 31, 46, 52] increase the vocabulary size but with limited accuracy. Motivated by the strong zero-shot abilities of VLMs, recent efforts focus on OVD. Finetuning-based methods [17, 18, 28, 48, 50] add detection heads onto pretrained VLMs and then finetune the detector with concepts of base categories. Such methods are simple but may forget the knowledge learned in the pretraining [2]. Distillation-based methods [7, 11, 40, 41] introduce additional distillation loss functions that force the output of a detector to be close to that of a VLM and thus avoid forgetting. However, since the distillation losses are not designed for the detection task, they may conflict with detection objectives due to gradient conflicts that are a common issue for multi-task models. Methods like Feng et al. [8], Gao et al. [9], Wu et al. [42], Zhao et al. [49] create pseudo labels (PL) of novel concepts as supervision, and do not require extra losses, which sidesteps both catastrophic forgetting and gradient conflicts. But these methods need handcrafted steps to generate high quality PLs, e.g. multiple runs of box regression [49], activation maps [9] from Grad-CAM [35], image retrieval [8], or multi-stage training [42]. In this work, we address the two challenges of using self-training for OVD and enable an efficient end-to-end pseudo labeling pipeline. Besides, we point out the noise due to the poor locations of PLs and introduce the SAF head to handle such noise.

Self-training for object detection. Weakly-supervised object detection methods [27, 33, 38, 39] usually explore online self-training by distilling the knowledge from the model itself. Recent semi-supervised object detection methods [19, 26, 45] adopt a teacher-student design, where the teacher is an exponential moving average (EMA) of the student. Self-training has been widely explored in the above fields but rarely in OVD. Unlike semi-supervised object detection, OVD encounters two challenges to use self-training, i.e., more noisy PLs, and larger changes in PLs' distributions. This work proposes the SAF head to address

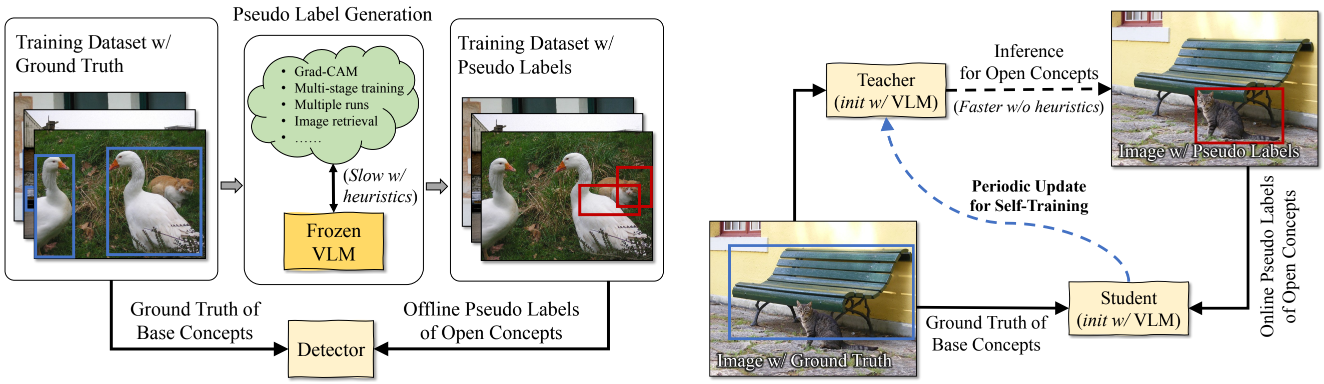

Figure 1. **Left:** Prior PL-based methods for OVD rely on handcrafted heuristics to leverage a frozen VLM for offline pseudo labels. This is usually inefficient and does not allow for improving PLs throughout training. **Right:** We customize self-training and finetune VLMs for OVD, which allows efficient on-the-fly computation of PLs that can be improved throughout training.

the noise, and adopts periodic updates to reduce the frequency of changes in PLs' distribution.

3. Approach

In open-vocabulary detection, an object detector is trained with bounding boxes and class labels of base categories $\mathcal{C}^B$. At inference, the detector is used for detecting objects of open concepts including $\mathcal{C}^B$ and novel categories $\mathcal{C}^N$, where $\mathcal{C}^B \cap \mathcal{C}^N = \varnothing$. To make such detection practical, recent studies [8, 11, 23, 48] adopt extra data (e.g., image-text pairs and image-level tags) and/or external VLMs. We follow their settings and leverage the pretrained CLIP [30] to build our OVD detector.

3.1. Adapting CLIP to OVD

In this section, we introduce how to adapt the ResNet-based CLIP into a Faster R-CNN [32] (C4) detector. For simplicity, we do not use FPN [25], but it can be incorporated with learnable gating from Flamingo [1]. Similarly, ViT-based CLIP models can be adapted to detectors following [22].

Region proposals from an external RPN. CLIP is pretrained via image-text alignment. As shown in Sect. 4.3, finetuning pretrained backbones to get region proposals decreases the performance, probably because such finetuning breaks the image-text alignment learned in the pretraining. To address this problem, F-VLM [18] freezes the pretrained backbone and only finetunes detection heads. But this solution limits the capacity of the detector. Unlike F-VLM, we follow the approach of Singh et al. [36], Zhong et al. [50] and employ an external RPN to generate region proposals, which is trained with ground truth boxes of base categories. Prior studies adopt RPN mainly to accelerate inference [36] or to improve region recognition [50]. By contrast, we aim to leverage the RPN to preserve the knowledge learned in the pretraining for better self-training.

Text embeddings as the classifier. For the i-th region proposal from the external RPN, we apply RoIAlign [13] on the 4th feature maps of CLIP's ResNet to get the proposal features. Then, the features are fed to the last ResNet block and the attention pooling of CLIP to get the region embedding r_i, which is later used for classification. Following prior studies [11, 23, 50], we convert a set of given concepts with prompt engineering into CLIP text embeddings, which act as classifiers, as shown in Fig. 2b. A fixed all-zero embedding is adopted for the background category. Assuming t_c is the embedding of the c-th category, the probability of r_i to be classified as the c-th category is,

$$p_{i,c} = \frac{\exp(\langle r_i, t_c \rangle / \tau)}{\sum_{j=0}^{C-1} \exp(\langle r_i, t_j \rangle / \tau)}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine distance, C denotes the vocabulary size, and τ denotes the temperature. In our detector, r_i may be further fed to a box refinement module to predict the box shift based on the region proposal. With the above adaptions, our initial detector gains zero-shot detection ability to some extent. Please refer to the supplement for quantitative evaluations.

3.2. Taming Self-Training

Although self-training has been widely explored in closed-set object detection, using it for OVD presents two challenges, i.e., noisy PLs and frequent changes of PLs' distributions. In the following, we describe our self-training pipeline and how to address the two challenges.

Self-training pipeline. Fig. 2a illustrates the pipeline of our self-training with a teacher-student manner. Training images are first fed into the RPN to obtain region proposals. Then, the teacher model runs inference on those proposals, where the resulting predictions with confidences above a threshold are selected as PLs. The student model adopts

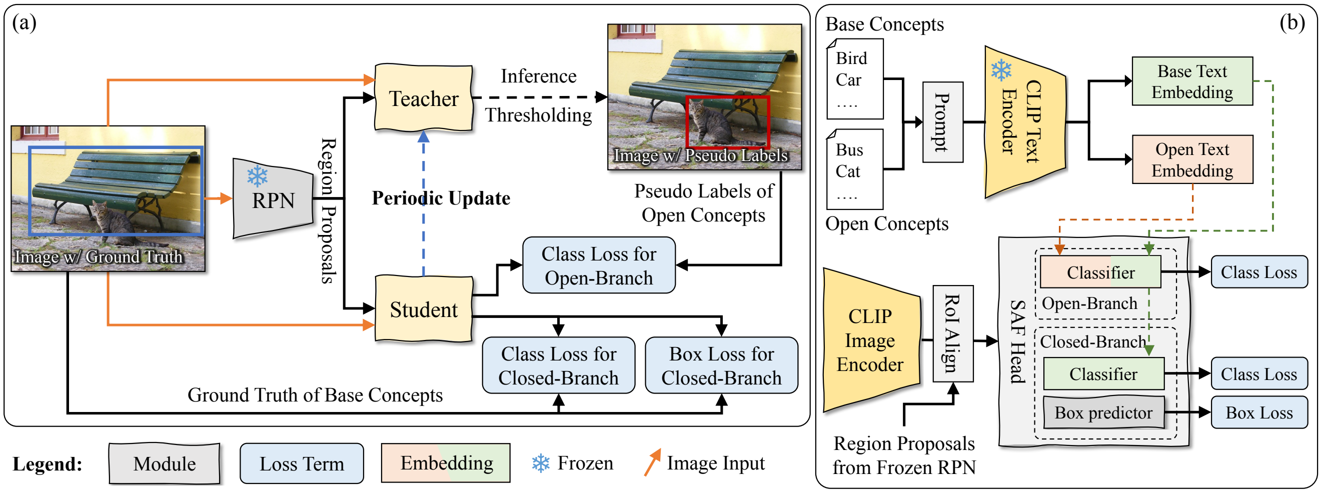

Figure 2. **(a) Pipeline of our self-training.** The teacher and the student are models of the same architecture. They are initialized with the same pretrained CLIP model. The teacher generates PLs that are used to train the student, and the student updates the teacher periodically. **(b) Structure of our detector.** The proposed SAF head is put on top of a CLIP image encoder. The open- and closed-branches take the text embeddings from a CLIP text encoder as classifier.

the same region proposals, and is supervised with both base ground truth and PLs generated by the teacher. The teacher is periodically updated with the parameters of the student model.

Handling noise in PLs. The noise of PLs from VLMs exists in both classification and localization. It is a common practice to filter PLs with classification confidence, but this only addresses noise in classification. To reduce the noise in locations of PLs, we exclude the noisy boxes of PLs from the training. That is, the classification loss is calculated on class labels of both ground truth and PLs, but the box regression loss is calculated only on ground truth boxes. Besides, to mitigate the impact of PLs on the performance of base categories, we propose SAF head and elaborate it in Sect. 3.3.

Reducing changes in PLs' distributions with periodic updates. The teacher model is updated with the student to enable self-training. The exponential moving average (EMA) is a widely used approach for object detection [19, 26, 45], which updates the teacher in every iteration. But we observed empirically that it does not benefit OVD (Table 4). We hypothesize that, unlike semi-supervised object detection, OVD has no ground truth for the target categories. Thus, the distribution of the target data (i.e., PLs) is fully determined by the teacher. The EMA update changes the PLs' distribution in each iteration and leads to a constantly shifting training target that has been shown hard to optimize (i.e., destabilizing the training) in deep Q-learning [29]. As a solution, we periodically update the teacher after a set number of iterations to maintain consistent distributions for PLs between updates. We call this strategy as the periodic update and show it outperforms the EMA update by a large margin in Sect. 4.3.

3.3. Split-and-Fusion (SAF) Head

Our SAF head first splits a detection head into two branches, i.e. "closed-branch" and "open-branch", with the goal to better handle noisy PLs during training. At inference, predictions from both heads are fused to boost the performance.

Splitting the detection head. The closed-branch follows a standard detection head with a classification module and a class-agnostic box refinement module. The former module classifies region proposals based on Eq. 1, and the latter one refines the proposal boxes for better locations. We train the closed-branch with boxes and class labels of ground truth for $\mathcal{C}^B$ using standard detection losses, which include a cross entropy loss for classification and a box regression loss for localization. Since no PLs are used to train the closed-branch, the noise of PLs is unlike to impact its performance on $\mathcal{C}^B$. Moreover, as shown in Gu et al. [11], Zareian et al. [48], box regression modules trained on $\mathcal{C}^B$ can generalize to novel categories that are unseen during training. Therefore, the closed-branch is able to provide generalized boxes, as well.

The open-branch only contains a classification module. It is trained using only the cross-entropy loss with class labels from both $\mathcal{C}^B$ and PLs, hence, learning broader concepts beyond $\mathcal{C}^B$. Unlike distillation losses [7, 11], all losses for the two branches are originally designed for detection and are unlikely to conflict with each other. When generating PLs, we use the classification scores from the open-branch and bounding boxes from the closed-branch. Boxes of PLs are not directly used in our losses but are involved to select foreground proposal candidates for the classification loss.

Fusing complementary predictions. The open- and closed-branches are trained in different ways and learn complementary knowledge. Therefore, we fuse their predictions with the geometric mean at inference time. Specifically, assuming $p_{i,c}^{\text{open}}$ and $p_{i,c}^{\text{closed}}$ are prediction scores of the open- and closed-branches, respectively, the final score is calculated as

$$p_{i,c}^{\text{fused}} = \begin{cases} (p_{i,c}^{\text{closed}})^{(1-\alpha)} \cdot (p_{i,c}^{\text{open}})^{\alpha}, & \text{if } i \in \mathcal{C}^B \\ (p_{i,c}^{\text{closed}})^{\alpha} \cdot (p_{i,c}^{\text{open}})^{(1-\alpha)}, & \text{if } i \in \mathcal{C}^N \end{cases} \tag{2}$$

where $\alpha \in [0, 1]$ balances the two branches. The indices i, c are the same as in Eq. 1. We keep a list of known base categories $\mathcal{C}^B$ and take other categories beyond the list as novel. It's important to note that fusion is a common strategy. For instance, F-VLM [18] employs score fusion to enhance the location quality of CLIP's predictions. In contrast, our approach focuses on fusing the complementary predictions from the two branches. The primary contribution of our method lies in learning what to fuse rather than the fusion process itself.

4. Experiments

4.1. Experiment Setup

Datasets. We conduct experiments on the two popular OVD benchmarks COCO-OVD and LVIS-OVD, based on COCO [24] and LVIS [12], respectively. (1) COCO-OVD: 65 out of 80 COCO categories are divided into 48 base categories and 17 novel categories. We report results on the validation set. (2) LVIS-OVD: 337 rare categories are used as the novel concepts and the remaining 866 categories (frequent and common) as the base concepts. We report results on the whole LVIS validation set.

Evaluation metrics. Following prior studies [11, 41, 48, 50], we report AP at IoU threshold of 0.5 for COCO-OVD. AP_{50}^{novel}, AP_{50}^{base} and AP_{50}^{all} are the metrics for the novel, the base and all (novel + base) categories, respectively. For LVIS-OVD, we report the mean AP averaged on IoUs from 0.5 to 0.95. AP_r, AP_c, AP_f and AP are the metrics for rare, common, frequent, and all categories.

Implementation details. We built our detector on Faster R-CNN with the CLIP version of ResNet (RN50-C4 or RN50x4-C4) as the backbone. Following prior studies [47, 49, 51], we assumed the novel concepts were available during training and took them as our open concepts, but no annotations of novel concepts are used. Our method was implemented on Detectron2 [43] and trained with 8 NVIDIA A6000 GPUs. We trained our models with the $1\times$ schedule (90k iterations) for COCO-OVD and the $2\times$ schedule for LVIS-OVD. The batch size was 16 with an initial learning rate of 0.002. The default data augmentation of Detectron2 was applied. Loss terms were equally weighted. By default, the teacher models were updated three times, usually together with the decreases of the learning rate.

4.2. Comparison with the Existing Methods

COCO-OVD. We compare our method with prior work on COCO in Table 1. With the same backbone and detector, SAS-Det outperforms the most recent method BARON [41] by 4.3 AP_{50}^{novel} on novel categories and 3.7 AP_{50}^{base} on base categories. Probably, BARON distills knowledge from VLMs, but distillation may lead to gradient conflicts and degrade the performance [49]. Compared to VLDet [23], SAS-Det gains an improvement of 7.9 AP_{50}^{base}. It is likely that VLDet employs noisy pseudo labels to train the detection head, impacting the detection of base classes in both classification and localization. In contrast, the two-branch design of our SAF head allows to train the closed-branch with just ground-truth of base classes and reduces the impact of the noise from pseudo labels. For more analysis on where our improvements come, please refer to Sect. 4.3 and Table 3. The first block of Table 1 provides results for methods with stronger backbones or detector architectures. When compared with those methods, SAS-Det still achieves the leading performance. Those results clearly demonstrate the effectiveness of SAS-Det.

LVIS-OVD. We provide the main results on LVIS in Table 2. When using ResNet50 as the backbone, SAS-Det achieves similar performance as the recent method DetPro [7]. DetPro proposes learnable prompts to generate better text embeddings as the classifier for OVD. Our method is orthogonal and can leverage DetPro's prompts for further improvement. The first block of Table 2 also shows that SAS-Det outperforms ViLD by a large margin on novel categories (indicated by AP_r) but gets lower AP_f for base categories. This is also observed in the recent method BARON [41]. The performance gap is probably due to the training setup. ViLD is trained for 8 times more iterations and adopts a strong data augmentation method [10], both of which benefit detection on base categories. In contrast, BARON and our approach adopt shorter training and standard data augmentations. When replacing the pretrained CLIP ResNet50 with ResNet50x4 [30] as the backbone, we improve AP_r by 8.2 from 20.9 to 29.1, which demonstrates SAS-Det scales up nicely with stronger pretrained VLM backbones.

4.3. Ablation Studies

In this section, the default *baseline* shares the same architecture and the training as our SAS-Det (with RN50-C4 as the backbone), except that the SAF head is replaced with a single detection head. The localization module of the detec-

Method	Training Setup	Backbone	Detector	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}
ViLD [11]	16×+LSJ	RN50-FPN	Faster R-CNN	27.6	59.5	51.2
VL-PLM [49]	1×+Default	RN50-FPN	Faster R-CNN	<u>32.3</u>	54.0	48.3
F-VLM [18]	0.5×+LSJ	RN50-FPN	Faster R-CNN	28.0	-	39.6
OV-DETR [47]	(Not Given)	RN50	Deform. DETR	29.4	61.0	52.7
CORA [42]	3×+Default	RN50	DAB-DETR	**35.1**	35.5	35.4
OVR-CNN [48]	(Not Given)	RN50-C4	Faster R-CNN	22.8	46.0	39.9
RegionCLIP [50]	1×+Default	RN50-C4	Faster R-CNN	26.8	54.8	47.5
Detic [51]	1×+Default	RN50-C4	Faster R-CNN	27.8	51.1	45.0
PB-OVD [9]	6×+Default	RN50-C4	Faster R-CNN	30.8	46.1	42.1
VLDet [23]	1×+LSJ	RN50-C4	Faster R-CNN	32.0	50.6	45.8
BARON [41]	1×+Default	RN50-C4	Faster R-CNN	<u>33.1</u>	54.8	49.1
SAS-Det (Ours)	1×+Default	RN50-C4	Faster R-CNN	**37.4**	58.5	53.0

Table 1. Comparison with recent methods on COCO-OVD. We group methods into two blocks. The first block contains methods using stronger backbones or detector architectures than ours. The second block contains models of the same scale as ours. Training setup indicates training iterations (N×) and data augmentations. Large Scale Jittering (LSJ) [10] is a stronger data augmentation than Detectron2's default.

Method	Training Setup	Backbone	Detector	AP_r	AP_c	AP_f	AP
ViLD [11]	16×+LSJ	RN50-FPN	Faster R-CNN	16.7	26.5	34.2	27.8
DetPro [7]	2×+Default	RN50-FPN	Faster R-CNN	<u>20.8</u>	27.8	32.4	28.4
F-VLM [18]	9×+LSJ	RN50-FPN	Faster R-CNN	18.6	-	-	24.2
BARON [41]	2×+Default	RN50-C4	Faster R-CNN	17.3	25.6	31.0	26.3
SAS-Det (Ours)	2×+Default	RN50-C4	Faster R-CNN	**20.9**	26.1	31.6	27.4
ViLD [11]	16×+LSJ	RN152-FPN	Faster R-CNN	19.8	27.1	34.5	28.7
OWL-ViT [28]	12×+LSJ	ViT-L/14	OWL-ViT	25.6	-	-	34.7
F-VLM [18]	9×+LSJ	RN50x4-FPN	Faster R-CNN	<u>26.3</u>	-	-	28.5
CORA [42]	3×+Default	RN50x4	DAB-DETR	22.2	-	-	-
SAS-Det (Ours)	2×+Default	RN50x4-C4	Faster R-CNN	**29.1**	32.4	36.8	33.5

Table 2. Comparison with recent methods on LVIS-OVD. We group methods based on the scale of backbones. Training setup contains training iterations (N×) and data augmentations. LSJ [10] is a stronger data augmentation than Detectron2's default.

tion head is trained with base ground truth boxes only, but the classification module is trained with class labels of both ground truth and PLs. Thus, the *baseline* provides a naive solution to exclude noisy pseudo boxes from the training. All evaluations are conducted on COCO.

External RPN. We leverage an external RPN to generate region proposals so that finetuning focuses on region-text alignment that is similar to the pretraining task. In this way, the detector is unlikely to forget the knowledge obtained in the pretraining. To validate the effectiveness of the external RPN, we follow F-VLM [18] to train a detector without an external RPN (See the supplement for more details). Table 3 compares *baseline* with the detector in the row of *(1)*. As shown, without the external RPN, the performance drops on both novel and base categories.

Removing boxes of PLs from training. Our *baseline* handles location noise of PLs' boxes by directly excluding them from the training. As shown in Table 3, *baseline* outperforms *(2)*'s detector. The only difference between them is the removal of pseudo boxes from training. This result clearly shows that simply removing the pseudo boxes is an effective way to deal with location noise.

Splitting the detection head. As shown in Table 3, compared to *baseline*, the open-branch of the proposed SAF head in *(5)* achieves similar performance on novel categories. This indicates that the split handles the noise in PLs' location as well as the naive solution of *baseline*. This is plausible because both solutions exclude the noisy boxes of PLs from training. Additionally, the open-branch gains 1.7 AP_{50}^{base} on base categories. Compared to *(3)*'s, the closed-branch in *(4)* gain improvements in terms of both AP_{50}^{novel} and AP_{50}^{base}. Note that *(4)*'s and *(3)*'s heads are trained with the same data. Based on those results, the split benefits OVD beyond handling the location noise. One advantage

Ablation	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}
Baseline	31.4	55.7	49.4
(1) No external RPN, train the backbone for region proposals	(-6.0) 25.4	53.4	46.1
(2) Noisy boxes of PLs as supervision for box regression	(-3.6) 27.8	55.2	48.0
(3) No PLs, train with base data only	(-8.2) 23.2	56.9	48.1
(4) W/ SAF head, use closed-branch's predictions ($p_{i,c}^{\text{closed}}$ in Eq. 2)	(-5.8) 25.6	57.9	49.5
(5) W/ SAF head, use open-branch's predictions ($p_{i,c}^{\text{open}}$ in Eq. 2)	(+0.5) 31.9	57.4	50.7
(6) W/ SAF head, use fused predictions ($p_{i,c}^{\text{fused}}$ in Eq. 2)	(+6.0) 37.4	58.5	53.0

Table 3. Ablation studies to analyze the effect of each component of SAS-Det on COCO-OVD. Results *(4)*, *(5)*, and *(6)* denote different outputs from the same model.

Update strategy	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}
Baseline (w/ periodic update)	31.4	55.7	49.4
(7) No teacher and no pseudo labels	(-8.2) 23.2	56.9	48.1
(8) Not update the teacher	(-1.8) 29.6	55.9	49.1
(9) Take the EMA of the student as the teacher	(-6.9) 24.5	53.9	46.2
(10) Replace the teacher with the student every iteration	(-23.4) 8.0	53.3	41.5

Table 4. Comparison of different strategies for updating the teacher in self-training on COCO-OVD.

is that, due to the split, the closed-branch is less likely to be influenced by the noise of PLs and learn how to better localize objects.

Fusing predictions. Our SAF head fuses predictions of the open-branch and the closed-branch at inference time. As shown in Table 3, *(6)*'s outperforms all the others by a large margin with the prediction fusion. Note that *(4)*'s, *(5)*'s and *(6)*'s refer to the different outputs of the same model. The improvement can be attributed to the fact that the two branches are trained on different sets of data and learn complementary knowledge. The closed-branch is trained with class labels and boxes of base categories. The open-branch is trained with class labels of more vocabularies including base and novel classes. Thus, the former learns more about how to localize objects and how to detect base classes. The latter learns more about how to detect new objects, which complements the former.

Update strategies for the teacher model. The teacher is updated with the student model to improve PLs during fine-tuning. We evaluated several strategies for updating the teacher model in Table 4 and summarize our findings as follows. First, the EMA update in *(9)*, which is shown effective and widely used in semi-supervised object detection [19, 26, 45], is just slightly better than training without PLs in *(7)*. It is worse than no update to the teacher (i.e. no self-training) in *(8)*. Second, if we replace the teacher with the student every iteration as *(10)*, AP_{50}^{novel} drops to 8.0 that is much lower than that number in *(7)*. These findings indicate that it's harmful to change the teacher frequently. We hypothesize that frequent updates change the distribution of PLs too often and makes the training unstable. Last but not least, by reducing the number of updates, our periodic update significantly outperforms the EMA update in OVD.

The number of updates to the teacher model. We trained our detectors with different numbers of updates to teacher models. Please see the supplement for how we distribute the updates. As shown in Table 5, too many updates, e.g., 8 or 4 updates, lead to performance drops mainly due to the following. First, similar as the aforementioned EMA update, too many updates change the distribution of PLs too often and make the training unstable. Second, the more updates, the earlier an update happens. However, the student model is not well trained at the early stage of the training and thus is not good enough to update the teacher. Table 5 shows that 2 and 3 updates achieve similar performance. But we set 3 updates as default to include as many updates as possible.

4.4. Further Analysis

Evaluation on the retained 15 COCO categories. Following prior studies [47, 49, 51], SAS-Det assumes novel concepts are available during training and achieves good performance. A natural question is if SAS-Det generalizes to alien concepts that are unknown before the evaluation. To answer the question, we follow the evaluation protocol of MEDet [5] where an OVD detector is evaluated on the whole COCO validation set with all 80 COCO concepts. AP_{50}^{retain}, AP_{50}^{novel} and AP_{50}^{coco} are reported as APs averaged on 15 retained, 17 novel, and all 80 COCO categories, respectively. As shown in Table 6, SAS-Det achieves leading performance by a clear margin, which indicates the good

# Update	AP_{50}^{novel}
8	27.6
4	30.6
3 (*baseline*)	31.4
2	**31.6**
1	30.9
0 (No update)	29.6

Table 5. Performance with varied numbers of updates to the teacher model.

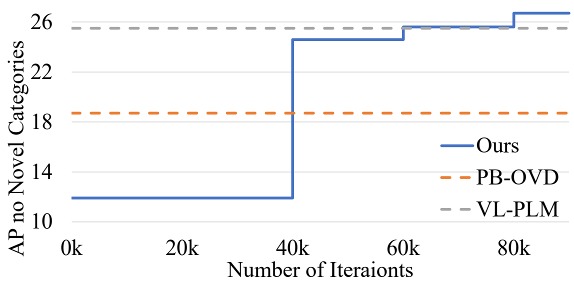

Figure 3. Quality of PLs during training.

Method	AP_{50}^{retain}	AP_{50}^{novel}	AP_{50}^{coco}
OVR-CNN [48]	11.5	22.9	38.1
Detic-80 [51]	11.5	27.3	38.3
MEDet [5]	18.6	32.6	42.4
Ours	**23.0**	**37.1**	**47.2**

Table 6. Generalization capability of OVD methods on retained 15 COCO categories. Text embeddings of 80 COCO categories are used as the classifier. Numbers are from MEDet [5].

Method	Time (s)	AP_{50}^{novel}
PB-OVD [9]	0.4848	18.7
VL-PLM [49]	0.4456	25.5
Ours	**0.1308**	**26.7**

Table 7. Mean time cost to get PLs per image. The quality of PLs on the validation set of COCO-OVD are provided for reference. We report the quality of our PLs after the last update.

Method	AP_{50}^{novel}	AP_{50}^{base}	AP_{50}^{all}
SAS-Det	37.4	58.5	53.0
Faster R-CNN + PLs	38.1	58.3	53.0

Table 8. Training Faster R-CNN with PLs from SAS-Det on COCO-OVD

generalization capability of the proposed method.

PL's quality during self-training. We compare our PLs and the prior state-of-the-art PLs from VL-PLM [49] on the COCO validation set in terms of PLs' quality. Following VL-PLM, we take AP_{50}^{novel} as the metric and evaluate our teacher models during training. As illustrated in Fig. 3, the quality of our initial PLs is not as good as VL-PLM's. But ours get close to VL-PLM's after two updates and become the better by the third update. This is because the teacher is initialized with CLIP, which is not pretrained for detection, and thus cannot provide high quality PLs. With updates, the teacher is equipped with the knowledge about detection and generates promising PLs.

Time cost of pseudo labeling. We run two major PL-based methods [9, 49] on our computational environment and compare them with our pseudo labeling in terms of the time cost in Table 7.

Reducing the time cost of external RPN. There is a simple solution to remove the extra computational cost of external RPN adopted by SAS-Det. We first generate pseudo labels with SAS-Det, and then follow VL-PLM [49] to train a Faster R-CNN based on ResNet50 for OVD without the external RPN ("Faster R-CNN + PLs"). As shown in Table 8, "Faster R-CNN + PLs" achieves slightly better or close performance as SAS-Det.

5. Conclusion

In this paper, we highlight two challenges associated with applying self-training to OVD: 1) noisy PLs from pretrained VLMs and 2) frequent changes of PLs' distributions. To address the challenges, we introduce a split-and-fusion (SAF) head and implement periodic updates. The SAF head splits a standard detection head into an open-branch and a closed-branch. The open-branch is trained with the class labels of ground truth and PLs. The closed-branch is trained with both class labels and boxes of ground truth. This approach effectively eliminates the location noise of PLs during training. Furthermore, the SAF head acquires complementary knowledge from different sets of training data, enabling fusion to enhance performance. The periodic updates reduce the number of updates to the teacher models, thereby decreasing the frequency of changes in PLs' distributions. We demonstrate that the proposed method SAS-Det is both efficient and effective. Our pseudo labeling is much faster than prior PL-based methods [9, 49]. SAS-Det also outperforms recent models on both COCO and LVIS for OVD.

Acknowledgments: This research project has been partially funded by research grants to Dimitris N. Metaxas through NSF: 2310966, 2235405, 2212301, 2003874, and FA9550-23-1-0417.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. *NeurIPS*, 35: 23716–23736, 2022. 3

[2] Relja Arandjelović, Alex Andonian, Arthur Mensch, Olivier J Hénaff, Jean-Baptiste Alayrac, and Andrew Zisserman. Three ways to improve feature alignment for open vocabulary detection, 2022. 2

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020. 1

[5] Peixian Chen, Kekai Sheng, Mengdan Zhang, Yunhang Shen, Ke Li, and Chunhua Shen. Open vocabulary object detection with proposal mining and prediction equalization, 2022. 7, 8

[6] Yuxiao Chen, Jianbo Yuan, Yu Tian, Shijie Geng, Xinyu Li, Ding Zhou, Dimitris N Metaxas, and Hongxia Yang. Revisiting multimodal representation in contrastive learning: from patch and token embeddings to finite discrete tokens. In *CVPR*, 2023. 2

[7] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14084–14093, 2022. 2, 4, 5, 6

[8] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, pages 701–717. Springer, 2022. 1, 2, 3

[9] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *ECCV*, pages 266–282. Springer, 2022. 1, 2, 6, 8

[10] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pages 2918–2928, 2021. 5, 6

[11] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *ICLR*, 2022. 1, 2, 3, 4, 5, 6

[12] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 5

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 3

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*, 2021. 2

[15] 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

[16] Zaid Khan, Vijay Kumar B.G., Xiang Yu, Samuel Schulter, Manmohan Chandraker, and Yun Fu. Single-stream multi-level alignment for vision-language pretraining. In *ECCV*, pages 735–751. Springer, 2022. 2

[17] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, pages 11144–11154, 2023. 2

[18] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023. 2, 3, 5, 6

[19] Gang Li, Xiang Li, Yujie Wang, Yichao Wu, Ding Liang, and Shanshan Zhang. Pseco: Pseudo labeling and consistency training for semi-supervised object detection. In *ECCV*, pages 457–472. Springer, 2022. 1, 2, 4, 7

[20] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 1, 2

[21] 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. In *CVPR*, pages 10965–10975, 2022. 2

[22] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, pages 280–296. Springer, 2022. 3

[23] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023. 1, 3, 5, 6

[24] 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*, 2014. 5

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 3

[26] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR*, 2021. 2, 4, 7

[27] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *CVPR*, pages 21404–21414, 2023. 2

[28] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. In *ECCV*. Springer, 2022. 2, 6

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 4

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 5

[31] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI*, pages 11932–11939, 2020. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 1, 3

[33] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *CVPR*, pages 10598–10607, 2020. 2

[34] Samuel Schulter, Yumin Suh, Konstantinos M Dafnis, Zhixing Zhang, Shiyu Zhao, Dimitris Metaxas, et al. Omnilabel: A challenging benchmark for language-based object detection. *ICCV*, 2023. 2

[35] 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. 2

[36] Bharat Singh, Hengduo Li, Abhishek Sharma, and Larry S Davis. R-FCN-3000 at 30fps: Decoupling detection and classification. In *CVPR*, pages 1081–1090, 2018. 3

[37] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. In *arXiv:2005.04757*, 2020. 1

[38] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, pages 2843–2851, 2017. 1, 2

[39] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *CVPR*, pages 2199–2208, 2019. 2

[40] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, pages 11186–11196, 2023. 2

[41] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, 2023. 2, 5, 6

[42] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. CORA: Adapting CLIP for Open-Vocabulary Detection with Region Prompting and Anchor Pre-Matching. In *CVPR*, 2023. 1, 2, 6

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

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020. 1

[45] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, pages 3060–3069, 2021. 1, 2, 4, 7

[46] Caixia Yan, Xiaojun Chang, Minnan Luo, Huan Liu, Xiaoqin Zhang, and Qinghua Zheng. Semantics-guided contrastive network for zero-shot object detection. *IEEE TPAMI*, 2022. 2

[47] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, pages 106–122. Springer, 2022. 5, 6, 7

[48] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-Vocabulary Object Detection Using Captions. In *CVPR*, 2021. 2, 3, 4, 5, 6, 8

[49] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, pages 159–175. Springer, 2022. 1, 2, 5, 6, 7, 8

[50] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-based Language-Image Pretraining. In *CVPR*, 2022. 2, 3, 5, 6

[51] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368. Springer, 2022. 5, 6, 7, 8

[52] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *CVPR*, pages 11693–11702, 2020. 2