

# DST-Det: Open-Vocabulary Object Detection via Dynamic Self-Training

Shilin Xu<sup>1\*</sup>, Xiangtai Li<sup>2,3\*†</sup>, Size Wu<sup>2</sup>, Wenwei Zhang<sup>2,3</sup>, Yunhai Tong<sup>1</sup>, Chen Change Loy<sup>2</sup>

<sup>1</sup>Peking University <sup>2</sup>S-Lab, Nanyang Technological University <sup>3</sup>Shanghai AI Laboratory

Code: <https://github.com/xushilin1/dst-det>

## Abstract

*Open-vocabulary object detection (OVOD) aims to detect the objects beyond the set of classes observed during training. This work introduces a straightforward and efficient strategy that utilizes pre-trained vision-language models (VLM), like CLIP, to identify potential novel classes through zero-shot classification. Previous methods use a class-agnostic region proposal network to detect object proposals and consider the proposals that do not match the ground truth as background. Unlike these methods, our method will select a subset of proposals that will be considered as background during the training. Then, we treat them as novel classes during training. We refer to this approach as the self-training strategy, which enhances recall and accuracy for novel classes without requiring extra annotations, datasets, and re-training. Compared to previous pseudo methods, our approach does not require re-training and offline labeling processing, which is more efficient and effective in one-shot training. Empirical evaluations on three datasets, including LVIS, V3Det, and COCO, demonstrate significant improvements over the baseline performance without incurring additional parameters or computational costs during inference. In addition, we also apply our method to various baselines. In particular, compared with the previous method, F-VLM, our method achieves a 1.7% improvement on the LVIS dataset. Combined with the recent method CLIPSelf, our method also achieves 46.7 novel class AP on COCO without introducing extra data for pertaining. We also achieve over 6.5% improvement over the F-VLM baseline in the recent challenging V3Det dataset. We release our code and models at <https://github.com/xushilin1/dst-det>.*

## 1. Introduction

Object detection is a fundamental task in computer vision, involving localization and recognition of objects within images. Previous detection methods [16, 36, 49] are

\*The first two authors contributed equally to this work. † Project Leader.

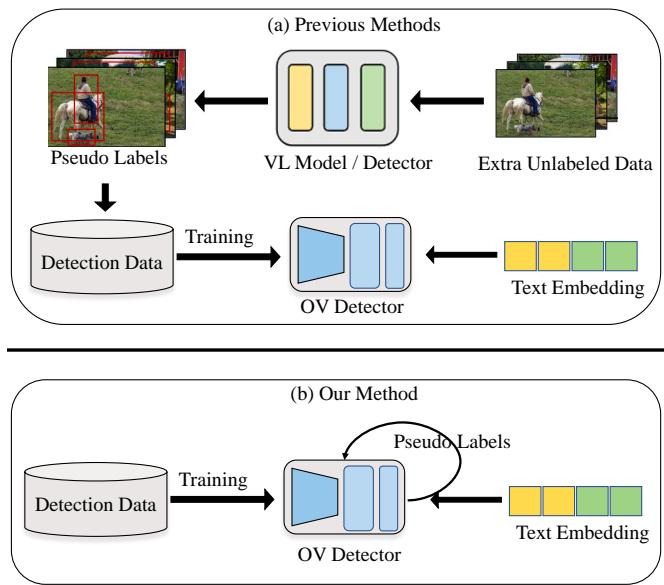


Figure 1. Compared with previous methods using the pseudo labels. **(a)**. Previous methods [12, 77, 78] usually obtain pseudo labels from extra unlabeled data. They first train a class-agnostic detector to generate object proposals and classify them using VLMs. **(b)**. Our DST-Det constructs an end-to-end pipeline, generates pseudo labels during training, and does not use any extra data.

limited to detecting only predefined categories learned during the training phase. This limitation results in a considerably smaller vocabulary compared to human cognition. Although directly extending the categories of large datasets [15] would be an ideal solution, it requires an overwhelming amount of human manual annotation. Recently, Open-Vocabulary Object Detection (OVOD) [14, 64, 73, 82] has emerged as a promising research direction to overcome the constraints of a fixed vocabulary and enable the detection of objects beyond predefined categories.

Typical solutions of OVOD rely on pre-trained VLMs [19, 44]. These VLMs have been trained on large-scale image-text pairs and possess strong zero-shot classification capability. Prior works [14, 73, 82] have attempted to leverage VLMs for OVOD by replacing learned classi-

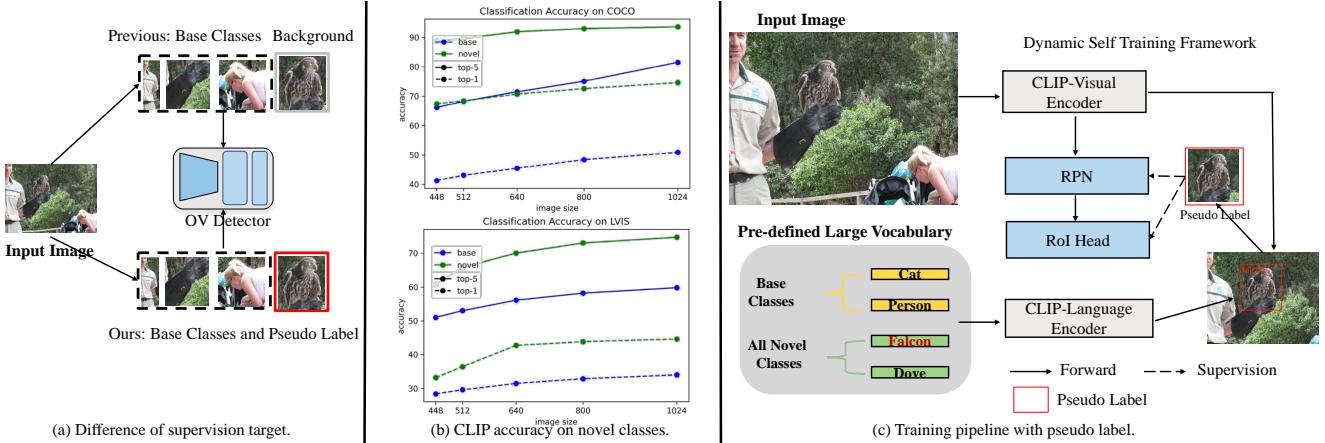


Figure 2. Illustration of our motivation and framework. (a). Our DST-Det incorporates novel class labels to supervise the detection head during training. (b). Experiments on OV-COCO and OV-LVIS using CLIP with ground truth box for zero-shot classification. We observe high top-1 and top-5 accuracy in classifying novel classes. (c). Illustration of our dynamic self-training pipeline with the pseudo labels.

fiers in traditional detectors with text embeddings derived from VLMs. Leveraging VLMs’ exceptional zero-shot classification ability enables the detector to assign objects to novel classes based on their semantic similarity to the embedded text representations. This is the default operation for all OVOD methods. Moreover, several recent approaches [14, 33, 66] aim to distill knowledge from VLMs by aligning individual region embeddings extracted from VLMs via diverse distillation loss designs. This alignment strategy transfers zero-shot capability from VLMs to the object detectors. Additionally, some studies [26, 68, 70, 71] attempt to build open-vocabulary detectors directly upon frozen visual foundation models. While these methods have shown impressive results in detecting new objects, there is a significant gap between training and testing for novel classes. This is because, during training, all novel classes are considered background classes. This issue has a particularly crucial effect on datasets with extensive vocabulary sizes. For example, a more recent challenging dataset, V3Det [59], contains over 10,000 classes. There are over 5,000 classes that are novel classes. However, during training, all these classes are treated as background.

In this work, we rethink the OVOD training pipeline and propose a new dynamic self-training approach by exploiting VLMs’ novel class knowledge. As shown in Fig. 2(a), most previous OVOD methods adopt the same training pipeline by using the base class annotations and treating novel objects as background. During testing, the object proposals are discovered by the class-agnostic region proposal network (RPN) and then classified using VLMs’ zero-shot ability. Thus, a conceptual gap exists between the training and testing phases when dealing with novel classes. Our approach aims to bridge this gap by classifying all object proposals using a large vocabulary size during both training and

testing phases rather than just during testing. Those proposals are categorized as novel classes that will be treated as pseudo-labels during training. Since our approach is carried out during the training stage, the pseudo examples are dynamic. Meanwhile, several previous methods also use pseudo labels. As shown in Fig. 1(a), previous methods [12, 78] first obtain pseudo labels from unlabeled data or image-caption datasets using VLMs, and then the detector is trained using generated pseudo labels (which contain the novel classes) and base class annotations. Our method provides an end-to-end training pipeline and generates pseudo labels dynamically during training. We list more features in Tab. 1. All methods need both base and novel class names to generate pseudo labels. However, our method is a plug-in-play approach that does not require extra data or trainable components. In addition, our method also does not require re-training with pseudo-labeled datasets.

To substantiate our motivation, we first conduct a toy experiment as illustrated in Fig. 2(b), where we calculate the top-1 and top-5 accuracy for both base classes and novel classes on LVIS and COCO datasets by using CLIP’s visual features and text embeddings. Specifically, we utilize the ground-truth boxes to obtain visual features from the feature maps of CLIP and calculate cosine similarity with text embeddings for zero-shot classification. Our results show that the top-5 accuracy for novel classes on LVIS and COCO datasets is already *remarkably* high enough to generate credible pseudo labels. This observation inspires us to consider directly using CLIP outputs as pseudo labels during the training phase. As shown in Fig. 2(a, c), we present the DST-Det (dynamic self-training detection) framework, which directly adopts the large vocabulary information provided by CLIP for training. The toy experiments also show the potential of our approach if there will

Table 1. Compared with existing methods using pseudo labels for OVD. Our method does not need extra data and extra components. Moreover, our method can be a plug-in method when VLMs are available.

Method	Extra Data	Extra Components	Plug-In Method	Dynamic Labeling	Need Vocabulary Names	Re-training
VL-PLM [78]	✓	✓	✗	✗	✓	✓
PB-OVD [12]	✓	✗	✗	✗	✓	✓
SAS-Det [77]	✗	✓	✗	✗	✓	✓
DST-Det (ours)	✗	✗	✓	✓	✓	✗

be strong VLMs [18] in the future.

To reduce the extra computation cost of the CLIP visual model, we let the object detection and the pseudo-labeling generation share the frozen CLIP visual backbone. This decision was supported by recent developments in exploiting frozen foundation models [26, 30, 68, 71]. Our DST-Det is based on the two-stage detector Mask R-CNN [16] and incorporates a dynamic pseudo-labeling module that discovers novel classes from negative proposals during training. Those proposals with high novel class scores can be considered foreground objects. By adopting simple threshold and scoring operations, we can effectively change some background proposals to the foreground according to the class scores. These simple designs do not incur any additional learnable parameters for novel class discovery, and the process is dynamic since region proposals vary in each iteration. In particular, we apply this operation to both the RPN and the Region-of-Interest Head (RoIHead) in the training stage: during RPN training, we force the discovered novel objects to be foreground objects; during RoIHead training, we add the pseudo labels directly to the classification target.

We show the effectiveness of DST-Det on the OV-LVIS [15], OV-V3Det [59] and OV-COCO [37] benchmarks. Our proposed method consistently outperforms existing state-of-the-art methods [26, 33, 66] on LVIS and V3Det without introducing extra parameters and inference costs. Specifically, DST-Det achieves 34.5% rare mask AP for novel categories on OV-LVIS. With the Faster-RCNN framework [49], our method achieves 13.5% novel classes mAP on V3Det, boosting previous methods by 6.8 % mAP. Moreover, our method improves considerably on the smaller vocabulary dataset COCO compared to the baseline method. In particular, with recent work [67] as a strong baseline, our method achieves the new state-of-the-art result with 46.7% AP on novel classes. We also provide detailed ablation studies and visual analyses, both of which validate the effectiveness of the DST-Det framework. Moreover, we further compare our method with two offline pseudo label methods [12, 78] under the fair comparison setting, where our method still achieves stronger results.

## 2. Related Work

**Close-Set Object Detection.** Modern object detection methods can be broadly categorized into one-stage and two-

stage. One-stage detectors [36, 40, 47, 57, 58, 75, 83, 84] directly classify and regress bounding boxes using a set of predefined anchors, where anchors can be defined as corners or center points. Recently, several works have adopted query-based approaches [5, 39, 54, 85] to replace the anchor design in previous works. The query-based approaches use a transformer decoder to decode each object directly without introducing hand-craft anchor boxes. Meanwhile, long-tail object detection aims to address class imbalance issues in object detection. To tackle this challenge, various methods have been proposed, such as data resampling [15, 38, 65], loss re-weighting [48, 55, 56, 63, 76], and decoupled training [32, 62]. However, all these methods cannot be generalized to novel categories. In the real world, we often need a detector to detect and recognize any objects under the large vocabulary scope rather than specific datasets. Our method focuses on the open-vocabulary detection setting.

**Open-Vocabulary Object Detection (OVOD).** This task extends the detector’s vocabulary at inference time, where the detector can recognize objects not encountered during the training. Recently, OVOD [2–4, 7, 8, 10–12, 14, 20, 29, 34, 41–43, 46, 50, 51, 53, 61, 66, 67, 72–74, 78, 79, 81, 82] has drawn increasing attention due to the emergence of vision-language models [19, 44]. On the one hand, several works [10, 14, 66, 72] attempt to improve OVOD performance by letting the object detector learn knowledge from advanced VLMs. For instance, ViLD [14] effectively distills the knowledge of CLIP into a Mask R-CNN [16]. Then, several following works adopt better knowledge distillation designs. DetPro [10] improves the distillation-based method using learnable category prompts, while BARON [66] proposes to lift the distillation from single regions to a bag of regions. Meanwhile, F-VLM [26] directly builds a two-stage detector upon frozen VLMs (CLIP) and trains the detector heads only. It fuses the predicted score and the CLIP score for open-vocabulary recognition. Moreover, several works [22, 23] aim for better VLMs pre-training for OVOD. Both types of open-vocabulary object detectors treat novel objects as background during training but target them as foreground objects in testing. This gap hinders the detector from discovering objects of novel categories. We aim to bridge this gap by leveraging stronger pre-trained vision-language models, enabling improved de-

tection performance for novel class detection without extra data or an extra re-training pipeline.

**Vision-Language Pre-training and Alignment.** Several works [19, 24, 27, 28, 44] study the pre-training of VLMs in cases of improving various downstream tasks, including recognition, captioning, and text generation. In particular, contrastive learning has achieved impressively aligned image and text representations by training VLMs on large-scale datasets, as demonstrated by several works [1, 19, 44]. For example, the representative work, CLIP [44], can perform zero-shot classification on ImageNet [9] without fine-tuning or re-training. Inspired by CLIP’s generalizability in visual recognition, various attempts have been made to adapt CLIP’s knowledge to dense prediction models such as image segmentation [13, 45, 70, 80] and object detection [14, 72] in the context of open-vocabulary recognition. Meanwhile, several works [52, 80] use CLIP to extract pseudo labels for dense prediction tasks. Our method also uses the knowledge of the pre-trained VLMs. Rather than using vision language alignment, our method effectively explores CLIP’s novel class-finding ability in the case of OVOD. In particular, CLIP models help discover the object of novel categories to bridge the conceptual gap between training and testing.

**Pseudo Labeling in Detection and Segmentation.** Pseudo-labeling methods [6, 25, 86] are well-explored in close-set segmentation and detection. Utilizing pre-trained models and unlabeled images achieves even better results than the original labeling models. For example, recent work SAM [25] scaled the unlabeled segmentation dataset using 1% annotated dataset. However, even with pseudo labels, via huge data training, SAN generated well in various scenes. For open-vocabulary detection and segmentation, several works [12, 17, 69, 77] also propose pseudo labels to improve OVD tasks. These approaches firstly pre-train a detector or a semgnenter using base class annotations. Then, they generate off-line base and novel labels with the aid of VLM. Next, the new models are jointly co-training with both base annotations and generated pseudo labels. However, these works require *extra VLM tuning and large caption datasets* for pre-labeling, which makes the pipeline complex. In contrast, our method is more straightforward and elegant, *without* extra pipelines or parameter tuning. In addition, compared with default pseudo-labeling methods [12, 17], shown in the experiment section, our method performs better on various datasets.

### 3. Method

**Overview.** This section begins by providing essential background information and discussing the baseline approaches employed in OVOD. We also introduce our baseline methods in detail. Subsequently, we introduce our proposed dynamic self-training method to address the identified issue.

Finally, we outline our approach’s training and inference details, building upon an existing detector architecture [16].

#### 3.1. Preliminaries and Baseline

**Problem Setting.** When given an image  $I$ , the object detector should output a set of boxes  $b_i$  and their corresponding class labels  $c_i$ . Here,  $b_i$  is a vector of length four representing the coordinates of the bounding box around the object,  $c_i$  is a scalar indicating the class label assigned to that object. During the training phase, the OVOD detector can only access the detection labels of base classes  $C_B$ . But it is required to detect objects belonging to the base classes  $C_B$  and the novel classes  $C_N$  at test time. The novel objects are unavailable during the training and are always treated as the background.

**Architecture Overview.** Most previous OVOD methods adopt a two-stage detector. Therefore, we take the Mask R-CNN [16] as an example to demonstrate how it can be adapted to the OVOD task by leveraging the textual information from pre-trained vision-language models. Mask R-CNN consists of two stages: a Region Proposal Network (RPN) and a Region-of-Interest Head (RoIHead). The RPN denoted as  $\Phi_{RPN}$  generates object proposals, and the RoI-Head denoted as  $\Phi_{RoI}$  refines the proposals’ locations and predicts their corresponding class labels. This process can be formulated as follows:

$$\{r_i\}_{i=1}^M = \Phi_{RoI} \circ \Phi_{RPN} \circ \Phi_{Enc}(I), \quad (1)$$

$$\{b_i, c_i\}_{i=1}^M = \{\Phi_{box}(r_i), \Phi_{cls}(r_i)\}_{i=1}^M, \quad (2)$$

where  $\Phi_{Enc}$  is an image encoder that maps the input image  $I$  to a series of multi-scale feature maps.  $M$  is the number of proposals generated by  $\Phi_{RPN}$ .  $\Phi_{RoI}$  extracts the region embedding  $r_i$  from the feature maps given a box proposal. We ignore the Feature Pyramid Network (FPN) [35] for brevity in this paper. Then the box regression network  $\Phi_{box}$  refines the coordinates of the bounding boxes, and the classification network  $\Phi_{cls}$  predicts the class label of the object within the bounding box. We use  $\circ$  to represent the cascade of different components.

The classification head in the close-set object detection is learnable in closed-vocabulary object detection and maps the region embedding into predefined classes. However, in the open-vocabulary scenario [14, 82], the classifier is substituted with text embeddings generated by pre-trained VLMs and is frozen during the training. The text embedding  $t_c$  for the  $c$ -th object category is generated by sending the category name into a CLIP text encoder using either a single template prompt, “a photo of category,” or multiple template prompts. And for a region embedding  $r$ , its classification score of  $c$ -th category is calculated as follows:

$$p_c = \frac{\exp(\tau \cdot \langle r, t_c \rangle)}{\sum_{i=0}^C \exp(\tau \cdot \langle r, t_i \rangle)} \quad (3)$$

where  $\langle \cdot, \cdot \rangle$  is the cosine similarity, and  $\tau$  is a learnable or fixed temperature to re-scale the value.

**Frozen CLIP as Backbone.** To reduce computation cost and strengthen the open-vocabulary ability of the object detector, we use the CLIP image encoder as the detector backbone in  $\Phi_{\text{Enc}}$ . We keep the parameters of our backbone fixed to preserve the vision-language alignment during training following F-VLM [26]. Then, the inference stage of the detector can benefit from both the detection score described in Eq. 3 and the CLIP score obtained by replacing the region embedding in Eq. 3 with the CLIP representation of the corresponding region proposal as depicted in Fig. 3(c). Specifically, we apply the RoIAlign function  $\Phi_{\text{RoI}}$  to the top-level feature map of  $\Phi_{\text{Enc}}$ . Then, the CLIP representations of the region proposals are obtained by pooling the outputs of the RoIAlign via the attention pooling module of the CLIP image encoder. Given a region proposal during the inference, the score of the  $c$ -th candidate category is obtained by merging the two types of scores via geometric mean:

$$s_c = \begin{cases} p_c^{1-\alpha} \cdot q_c^\alpha & \text{if } i \in C_B \\ p_c^{1-\beta} \cdot q_c^\beta & \text{if } i \in C_N \end{cases} \quad (4)$$

where  $q_c$  is the CLIP score,  $\alpha, \beta \in [0, 1]$  control the weights of the CLIP scores for base and novel classes, and  $C_B$  and  $C_N$  represent the base classes and novel classes, respectively.

### 3.2. DST-Det: Dynamic Self-Training For OVOD

**Motivation of DST-Det.** Previous works have addressed the problem of open-ended classification in OVOD by utilizing the text embeddings from pre-trained VLMs (e.g., CLIP). However, a conceptual gap exists between training and testing for novel classes. As shown in Fig. 2, annotations for novel objects are considered background during training. However, during the test phase, they are expected to be detected as foreground and classified into a specific novel class based on their detection score. A new training strategy is needed to bridge the gap between the training and test phases by utilizing CLIP’s outputs as pseudo labels for novel classes. In the toy experiments depicted in Fig. 2(b), we treat the ground-truth bounding boxes as region proposals to get the CLIP score for each region proposal and use the CLIP scores to verify the zero-shot classification ability of CLIP on OV-COCO and OV-LVIS datasets. How to get the CLIP score has been illustrated in Sec. 3.1 and Fig. 3(c). The results indicate that the top-5 accuracy suffices for discovering novel classes. This motivates us to consider using CLIP outputs as pseudo labels during training, given the RPN proposals and a large pre-defined vocabulary.

**Pseudo-Labeling Module (PLM).** As shown in Fig. 3 (b), we categorize proposals into two types: *positive* and *negative*.

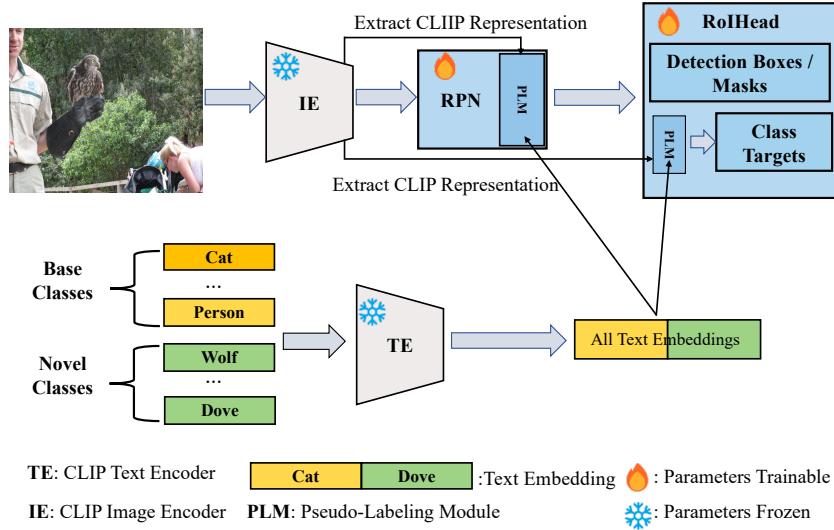
*Positive* proposals refer to those having large overlaps with ground truth object bounding boxes and are used to learn box regression and classification, which are supervised by base class annotations. On the other hand, the negative proposals have a low overlap with the ground-truth boxes but potentially contain objects of novel classes. To avoid treating region proposals of potential novel classes as background contents, our PLM selects negative proposals whose region embeddings have high similarity scores with the text embeddings of novel classes as pseudo labels.

We believe that negative proposals, which have a low overlap with the ground truth boxes, may contain objects belonging to novel classes. Therefore, we introduce the Pseudo-Labeling Module (PLM) to avoid treating these proposals as background. As shown in Fig. 3(b) and Fig. 3(c), we first extract the CLIP representations of the negative proposals. We treat the negative proposals as regions of interest and use the RoIAlign [16] operation to extract region embeddings from the top-level of the backbone features (CLIP backbone). These region embeddings are mapped to the representation space of CLIP text embeddings using a fixed attention pooling layer [44]. Then, we calculate the cosine similarity scores between the region embeddings and the text embeddings of the object class names. After obtaining the similarity scores, we filter out those proposals classified as base classes or background classes and those with CLIP scores lower than a threshold. The remaining few proposals can be identified as novel objects, and we randomly select K proposals as pseudo labels and change the classification target of these proposals during training. The targets of the positive proposals remain unchanged in this process.

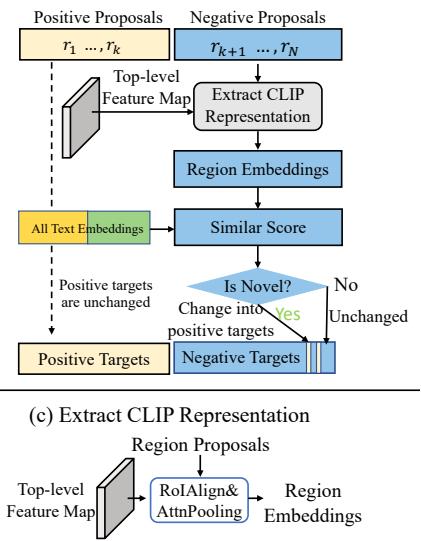
**Deployment on RPN and RoIHead.** Due to the dense anchor head, the RPN stage typically produces numerous negative proposals. To reduce the PLM’s computation cost, we leverage Non-Maximum Suppression (NMS) to eliminate redundant proposals and limit the number of negative proposals to a fixed amount, such as 1000. In contrast, there are fewer negative proposals in the RoIHead stage compared to the RPN stage. As a result, we send all negative proposals to the PLM during the RoIHead stage. The PLM will select some negative proposals as pseudo-labels according to our strategy and change their classification target. During the RPN stage, we convert the classification target of selected negative proposals from negative to positive. For the classification branch of RoIHead, we change the classification target of selected negative proposals from negative to pseudo labels generated by the PLM. Note that we only apply the pseudo labels produced by PLM to classification losses. The box regression and mask losses remain unchanged for negative proposals, as we cannot create a pseudo box or mask just using VLMs. The detailed process is shown in Fig. 3 (a).

**Discussion.** The proposed PLM is only used during the

(a) The meta architecture of DST-Det



(b) The Pseudo-Labeling Module (PLM)



(c) Extract CLIP Representation

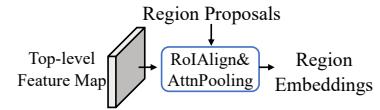


Figure 3. Illustration of DST framework. (a) The meta-architecture of DST-Det includes the proposed pseudo-labeling module (PLM), which is integrated into two stages of the detector: RPN and RoIHead. (b) The proposed pseudo-labeling module (PLM). During training, PLM takes the top-level feature map from the image encoder and text embedding of object classes as input and generates the pseudo labels for the RPN and the RoIHead. (c) The process of extracting CLIP representation for region proposals. The ROIAlign operation is applied to the top-level feature map, the output of which is then pooled by the Attention Pooling layer (AttnPooling) of the CLIP image encoder.

training without introducing extra parameters and computation costs during the inference. The pseudo-labeling process requires a vocabulary that contains potential novel object classes. We can either obtain the novel classes from the detection datasets following Detic [82] or collect a wider range of object classes from external sources, e.g., the classes defined in image classification datasets [9].

Moreover, the PLM training is single-stage and end-to-end. We list the detailed progress in Algo. 1 table. In particular, given the negative proposal set  $P_N$  and corresponding score  $S$ , our algorithm dynamically selects the potential candidates as novel foreground classes. By only changing the classification loss in both RPN and RoIHeads, our method does not bring any extra trainable parameters and computation costs. In addition, our method can be applied to various RPN-based approaches, which makes it easy to transfer.

### 3.3. Training and Inference

The training losses adhere to the default settings of Mask R-CNN [16], which comprises three primary losses: classification loss, box regression loss, and mask segmentation loss. Specifically, only the classification loss will be changed based on the pseudo labels, while other losses will remain unmodified. It is important to note that the generated labels are utilized for recognition purposes rather than localization. Therefore, the final loss is expressed

as:  $L = \lambda_{ps\_cls} L_{pl\_cls} + \lambda_{box} L_{box} + \lambda_{mask} L_{mask}$ .  $\lambda$  denotes the weight assigned to each loss, and  $L_{pl\_cls}$  is our modified pseudo label classification loss. Specifically, we set a weight of 0.9 for the background class and 1.0 for all other classes. For inference, we adopt the same score fusion pipeline (Equ. 4) as stated in Sec. 3.1.

## 4. Experiments

In this section, we will first introduce the experiment settings and baselines on three different benchmarks. Then, we report results on three different datasets. We also include the comparison experiment results. Next, we explore the extensive ablation studies of our DST-Det. Finally, we further present visualization results and analysis.

### 4.1. Experiment Settings

**Datasets and Metrics.** We carry out experiments on three detection datasets, including OV-LVIS [14], OV-COCO [73], and recent challenging OV-V3Det [59]. For LVIS, we adopt the settings proposed by ViLD [14] and mainly report the  $AP_r$ , representing the AP specifically for rare categories. We report box AP at a threshold of 0.5 for COCO and mean AP at threshold  $0.5 \sim 0.95$  for V3Det. All the datasets used in this paper are available online. COCO<sup>1</sup>,

<sup>1</sup><https://cocodataset.org>

---

**Algorithm 1** Dynamic Self-Pseudo-Labeling Training Process

---

**Input:** negative proposal set  $P_N$ , score  $S$  for each  $p \in P_N$ , top-level features  $F$  from image encoder, text embeddings  $t$  for all categories, classification label  $L_{cls}$

**Output:** pseudo classification label  $L'_{cls}$

```
1: candidates = []
2: for  $i, p$  in enumerate( $P_N$ ) do
3:   if  $S_i > 0$  then
4:      $r_i \leftarrow \text{RoIAlign}(F, p)$  /* region embeddings */
5:      $s_i \leftarrow \text{MatMul}(r_i, t)$  /* calculate potential novel classes scores */
6:     if  $\max(s_i) > 0.8$  and  $\text{argmax}(s_i) \in \text{novel classes}$  then /* find the novel objects */
7:       candidates  $\leftarrow i$  /* put the novel objects in the training candidates. */
8:   Keep selected_idx  $\leftarrow$  randomly select K RoIs from candidates /* select the previous candidates */
9:   Update  $L'_{cls}[\text{selected\_idx}] \leftarrow \text{pseudo novel classes}$  /* classification loss */
10:  Use  $L'_{cls}[\text{selected\_idx}]$  to train the RPN and RoIHead for the classification branch. /* update both RPN and RoIHead for classification. */
11:  Keep the detection box loss and segmentation mask loss (For Mask-RCNN baseline) unchanged.
```

---

LVIS<sup>2</sup>, and V3Det<sup>3</sup> can be downloaded from their official website accordingly.

**Implementation Details.** We follow the settings of F-VLM [26] for a fair comparison. We use the Mask R-CNN [16] framework with feature pyramid network [35] as our detector. All the class names are transferred into CLIP text embedding, following [14]. For the “background” category, we input the word “background” into the multiple templates prompt and get a fixed text embedding from the CLIP text encoder. We only use the base boxes for training and all class names as of the known vocabulary as the pre-knowledge of PLM. For the final results on LVIS, we train the model for 59 epochs. For ablation studies, we train the model for 14.7 or 7.4 epochs. For the COCO dataset, we follow the previous works [26]. For the V3Det dataset, we adopt the default setting of the origin baselines [59]. We have released all the source code and models for further studies.

**Dataset and Training Details on OV-LVIS.** LVIS [15] has a large vocabulary and a long-tailed distribution of object instances. The LVIS dataset contains bounding box and instance labels for 1203 classes across 100k images from the COCO dataset. The classes are categorized into three sets based on their occurrence in training images – rare, common, and frequent. And for open vocabulary object detection, we adopt the settings proposed by ViLD [14]. In this approach, annotations that belong to common and frequent categories are categorized as base categories. On the other hand, annotations belonging to rare categories are treated as novel categories. We use the SGD optimizer with a learning rate of 0.36 and a weight decay of 1e-4. We train our model for 46.1k iterations with a batch size of 256.

<sup>2</sup><https://www.lvisdataset.org/>

<sup>3</sup><https://v3det.openxlab.org.cn/>

**Dataset and Training Details on OV-COCO.** We follow the setting of ovr-cnn [73] and split COCO2017 into 48 base classes and 17 novel classes. We train our model in this setting for 11.25k interactions with a batch size of 64. Other experiment settings are the same as OV-LVIS.

**Dataset and Training Details on OV-V3Det.** OV-V3Det [59] is a vast vocabulary visual detection dataset. It contains extremely large categories, which consist of 13,029 categories on real-world images. The train split of V3Det 1,361,181 objects in 184,523 images, the val split has 178,475 objects in 30,000 images, and the test split has 190,144 objects in 30,000 images. For the open vocabulary object detection setting, V3Det randomly samples 6,501 categories as base classes  $C_{base}$  and the remaining 6,528 categories as the novel classes  $C_{novel}$ . For training a detector for V3Det, we use the SGD optimizer with a weight decay of 1e-4, a momentum of 0.9, and a learning rate of 0.02. We train our model for 2x with a batch size of 32.

## 4.2. Main Results

This section mainly reports the results of three open-vocabulary detection datasets (See Tab. 3, Tab. 2, and Tab. 4). Considering that the training data are different, we will further include a fair comparison with the previous pseudo-label methods (See Tab. 5).

**Results on LVIS OVOD.** Tab. 2 shows the results of our approaches with other state-of-the-art approaches on the LVIS dataset. Due to the limited computing power, we use a smaller batch size (64 vs. 256) and shorter training schedule (59 epochs vs. 118 epochs) to build our baseline, which results in 32.0 % mAP, which is lower than F-VLM [26]. After applying our methods, we obtain 34.5% mAP on rare classes, about 1.7% mAP higher than F-VLM [26], without introducing any new parameters and cost during infer-

Table 2. Results on OV-LVIS benchmark

Method	Backbone	mAP <sub>r</sub>
ViLD [14]	RN50	16.6
OV-DETR [72]	RN50	17.4
DetPro [10]	RN50	19.8
VL-PLM [78]	RN50	17.2
OC-OVD [46]	RN50	21.1
OADP [60]	RN50	21.7
RegionCLIP [79]	RN50x4	22.0
CORA [68]	RN50x4	22.2
BARON [66]	RN50	22.6
VLDet [33]	SwinB	26.3
EdaDet [50]	RN50	23.7
MultiModal [21]	RN50	27.0
CORA+ [68]	RN50x4	28.1
F-VLM [26]	RN50x64	32.8
Detic [82]	SwinB	33.8
RO-ViT [23]	ViT-H/16	34.1
CLIPSelf [67]	ViT-B/16	25.3
Baseline	RN50x16	26.3
DST-Det	RN50x16	28.4 (2.1 ↑)
Baseline	RN50x64	32.0
DST-Det	RN50x64	34.5 (2.5 ↑)
DST-Det+CLIPSelf [67]	ViT-B/16	26.2 (0.9 ↑)

Table 3. Results on OV-COCO benchmark

Method	Backbone	AP <sub>50</sub> <sup>novel</sup>
OV-RCNN [73]	RN50	17.5
RegionCLIP [79]	RN50	26.8
ViLD [14]	RN50	27.6
Detic [82]	RN50	27.8
PB-OVD [12]	RN50	30.8
VL-PLM [78]	RN50	34.4
F-VLM [26]	RN50	28.0
OV-DETR [72]	RN50	29.4
VLDet [33]	RN50	32.0
RO-ViT [23]	ViT-L/16	33.0
RegionCLIP [79]	RN50x4	39.3
CLIPSelf [67]	ViT-B/16	37.6
CLIPSelf [67]	ViT-L/14	44.3
Baseline	RN50x64	27.4
DST-Det	RN50x64	33.8 (6.4 ↑)
DST-Det + CLIPSelf [67]	ViT-B/16	41.3 (4.7 ↑)
DST-Det + CLIPSelf [67]	ViT-L/14	46.7 (2.4 ↑)

ence. Compared with other stronger baselines, including VLDet [33] and Detic [82], our method does not use any extra data and achieves about 0.7-8.2% mAP gains. We also find consistent gains over different VLM baselines. We further show the generalization ability of our methods on re-

Table 4. Results on OV-V3Det benchmark

Method	Backbone	AP <sup>novel</sup>
Detic [82]	RN50	6.7
RegionClip [79]	RN50	3.1
Baseline	RN50	3.9
DST-Det	RN50	7.2 (3.3 ↑)
Baseline	RN50x64	7.0
DST-Det	RN50x64	13.5 (6.5 ↑)

Table 5. Comparison with Previous Pseudo Labeling Methods. We adopt RN50x64 backbone as strong baselines for each dataset.

Method	COCO	LVIS	V3Det
Baseline	27.4	32.0	7.0
Method [12]	29.2	32.4	7.4
Method [78]	31.2	32.5	8.1
Ours	33.8	32.5	13.5

cent work, CLIPSelf [67] in the last row of Tab. 2. Without any further introduced parameters and GFlops, our method still improves CLIPSelf by 0.6% mAP gains. In addition, our method also outperforms previous pseudo method VL-PLM [78] by a large margin.

**Results on COCO OVOD.** Tab. 3 shows the results of our approaches with other state-of-the-art approaches on the COCO dataset. We achieve 27.4% novel box AP when adopting the frozen Mask R-CNN baseline. After using our offline refinement for the self-training module, our method can achieve a 6.4% improvement on novel box AP. Both experiment results indicate the effectiveness and generalization ability of our approach. Moreover, we also verify our method on a stronger detector, CLIPSelf [67]. With a stronger VLM as the backbone and a stronger detector as the baseline, our method still achieves 2.4-4.7% improvement on novel classes, which results in a new state-of-the-art result using **only** COCO datasets. This indicates our proposed method is a general and plug-in component to improve VLM-based OVOD methods.

**Results on V3Det OVOD.** V3Det is a more challenging dataset with a larger vocabulary size than both LVIS and COCO. As shown in Tab. 4, our methods achieve new state-of-the-art results on different backbones. In particular, with RN50x64, our method achieves 13.5 % mAP on novel classes, significantly outperforming previous STOA by 6.8 %. Moreover, with different backbones, our methods can still improve the strong baseline via 3.3-6.5%.

**Comparison to Previous Pseudo Labeling Method.** Although in Tab. 3 and Tab. 2 our method outperform previous pseudo labeling methods, both the training data and baseline are different. Thus, in Tab. 5, we compare the previous method under a fair setting, including using the same

Table 6. Ablation studies and comparative analysis on LVIS 1.0 dataset. PLM<sup>1</sup> means PLM in RPN and PLM<sup>2</sup> means PLM in RoI head. By default, we add two PLMs. We mainly report AP<sub>r</sub> for reference. AP<sub>all</sub> is also used for all classes. All methods use the strong RN50x64 backbone on LVIS. For results on COCO, we report results using RN50.

(a) Effectiveness of PLM.				(b) Loss Choices of PLM.			(c) Training Schedule. e: epoch.		
Baseline	PLM <sup>1</sup>	PLM <sup>2</sup>	AP <sub>r</sub>	Setting	AP <sub>r</sub>	AP <sub>all</sub>	Setting	Schedule	AP <sub>r</sub>
✓	-	-	28.3	Baseline	28.3	30.2	Baseline	7.4 e	28.3
✓	✓	-	30.4	box loss	10.3	13.6	w DST	7.4 e	32.2
✓	-	✓	29.5	class + box loss	20.3	25.2	Baseline	14.7 e	31.2
✓	✓	✓	31.8	class loss	31.8	31.5	w DST	14.7 e	33.4

(d) More Design Choices of PLM.				(e) Supervision Vocabulary During Training.			
CLIP score	K	AP <sub>r</sub>	Source	AP <sub>r</sub>	AR <sub>all</sub>	AP <sub>r</sub>	AR <sub>all</sub>
0.5	4	23.3	base names only	28.3	30.2		
0.8	4	31.8	using LVIS rare	31.8	31.5		
0.4	4	13.2	using ImageNet	31.5	31.2		
0.8	15	25.6					

Table 7. GFLOps and Parameter Analysis. Our method **does not** bring extra flops or parameters during inference.

	Method	Backbone	GFLOps	Learnable Param.	Frozen Param.
Train	Baseline	CLIP RN50	343G	22.9M	38.3M
		CLIP RN50x64	1565G	23.9M	420M
	Ours	CLIP RN50	561G	22.9M	38.3M
		CLIP RN50x64	4961G	23.9M	420M
Inference	Baseline	CLIP RN50	813G	22.9M	38.3M
		CLIP RN50x64	8157G	23.9M	420M
	Ours	CLIP RN50	813G	22.9M	38.3M
		CLIP RN50x64	8157G	23.9M	420M

detector in the same codebase. For the table, compared with previous pseudo labeling methods [12, 78], we find our approach works best, even without re-training. Moreover, we find our approach is more effective on larger vocabulary datasets, including LVIS and V3Det. This is because, in those datasets, rare classes are treated as background during the training. Thus, a simple trained detector cannot generalize well without extra text-image datasets.

### 4.3. Ablation Study and Analysis

In this section, we conduct detailed ablation studies and analyses on our DST-Det.

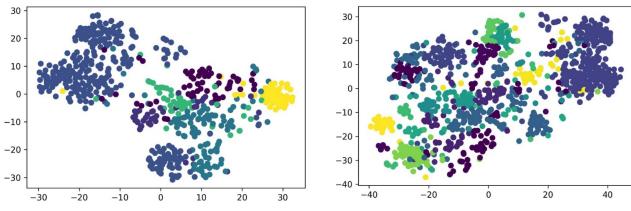
**Effectiveness of PLM.** In Tab. 6(a), we first verify the effectiveness of PLM. Adding PLM in RPN obtains 1.9 % mAP improvements while inserting PLM in RoI heads leads to 1.2% gains. This indicates the novel classes are more sensitive to RPN. As shown in the last row, combining both yields better results, confirming the orthogonal effect of the two classification losses. As a result, the PLM results in a notable improvement of over 3.0%.

**Ablation on Loss Design in PLMs.** In PLM, we only adopt the classification loss of generated labels and ignore the box

loss. This is motivated by the fact that most generated proposals are inaccurate, adversely affecting the localization ability developed using normal base annotations. To validate this assertion, we add the box loss into the PLM. As shown in Tab. 6(b), we observe a significant drop in both AP<sub>r</sub> and AP<sub>all</sub>. Furthermore, our objective is to enable the detector to recognize the novel objects or part of the novel objects. As shown in Fig. 4, repetitive proposals still persist even with filtering operations in PLMs.

**Ablation on Various Schedules.** In Tab. 6(c), we verify the effectiveness of longer schedule training. After extending the schedule into 14.7 epochs, we still observe over 2.0% improvements. Finally, we extend the training schedule to 59 epochs and still obtain 2.5% improvements, as shown in Tab. 2. This indicates our method can be scaled up with longer training and stronger baselines.

**Ablation on Effect of Pseudo Score and Pseudo Label Number.** We adopt CLIP score and training samples K to select the possible proposals, since most are background noises. As shown in Tab. 6(d), decreasing the CLIP score and increasing training samples lead to inferior performance. Decreasing the score may involve more irrele-



(a), Novel proposals embeddings.  
K=4 (left), K=20 (right)



(b), Visual improvements over strong baseline.  
Baseline (left), Ours (right)

Figure 4. Visual Analysis of DST framework. (a), We present a t-SNE analysis on the novel region embeddings during training. Different colors represent different classes. We find that using fewer training samples works well. (b), We show visual improvements over the strong baseline. Our method can detect and segment novel classes, as shown on the right side of the data pair.

vant objects or context. Increasing training examples may result in more repeated and occluded objects. Moreover, we also visualize the embeddings of novel proposals during the training with different  $K$  in Fig. 4(a). With more training examples, the novel class embeddings become less discriminative, contributing to increased noise. Thus, we set  $K$  to 4 and the CLIP score to 0.8 by default.

**Ablation on Source of Vocabulary.** We also verify the vocabulary source for novel classes in Tab. 6(e). We experiment with two types of sources: the classes (including rare classes) defined in LVIS [15] and the classes defined in ImageNet [9]. As shown in Tab. 6(e), our approach achieves considerable performance even when we *do not* have access to the novel classes in the test dataset and use an external source of potential novel classes, i.e., ImageNet. The source of large vocabulary maybe more important when adopting multi-dataset co-training.

**GFLops and Parameter Analysis.** In Tab. 7, we list the GFLops and number of parameters during training and

inference. Our method uses a frozen backbone and a learnable detection head, and the parameters mainly come from the backbone. When using a large backbone, CLIP RN50x64 [44], the number of learnable parameters accounts for only one-twentieth of the total number. During training, our method will get 512 proposals per image and obtain its embeddings through RoIAlign [16] and *AttentionPool* operation of CLIP. Compared to the baseline model, our method contributes a significant amount of GFlops from *AttentionPool* operation. During inference, we use the same inference pipeline in the baseline method and use 1000 proposals per image. A larger number of proposals leads to higher GFlops.

#### 4.4. Visualization, Error Analysis, and Future Work

This section mainly presents visual examples of visual improvements, pseudo-label examples, visual analysis, and failure case analysis. In addition, we also point out the impact of our work, limitations, and future work.

**Visual Improvements Analysis.** In Fig. 4 (b), we present several visual improvements over a strong baseline (32.0 AP<sub>r</sub>) on LVIS. Compared to the baseline, our method demonstrates effective detection and segmentation of novel classes, such as soup and beer bottles. In Fig. 6, we present more visual examples of the LVIS datasets. These classes are rare and do not have any annotations during the training, which proves the effectiveness of finding novel and rare classes.

**Qualitative Novel Class Example in DST.** We further visualize the generated novel examples in Fig. 4. Most visual examples exhibit satisfactory quality after the filtering operation in PLMs. Compared with previous methods, which treat the novel classes as background, our method directly forces the detector to train the classification head with the assistance of VLM. Despite overlapping or repetitive proposals, we can successfully train the detector by modifying only the classification loss within the RPN and RoI heads.

**Failure Cases Visualization.** In Fig. 7, we present several failure examples on LVIS. Although our method introduced the pseudo-labeling module for novel classes during training, many novel class objects cannot be detected or classified correctly. We argue that adopting stronger VLMs can solve this issue.

**Pseudo Labels Visualization.** This section presents additional visualization results. Specifically, we present the pseudo labels generated by our pseudo-labeling module in Fig. 5. The visualization results are alongside the ground truth annotations for comparison. Our pseudo labels can successfully recall the novel class annotations not seen during training.

**Limitation and Future Work.** Our proposed approach leverages a better VLM model to generate high-quality



Figure 5. Pseudo Label Visual Examples. Left: We visualize the class-agnostic ground truth bounding boxes. The green boxes represent the ground truth of base classes and will be used as foreground supervision, while the red boxes represent the ground truth of possible novel classes that are not allowed during training. Right: The red boxes represent the pseudo labels we selected from negative proposals.

pseudo labels, where the zero-shot ability of VLMs affects the quality of novel class labels. However, with more strong VLMs in the future [31, 44], our method is more feasible for large vocabulary applications. Another direction is to explore our framework under multi-dataset detection using one model with a large vocabulary size, which will fully utilize the potential of VLM’s vocabulary scope to generate more accurate labels.

**Board Impact.** Our method designs the first dynamic self-training on OVOD via mining the self-knowledge from the VLM model and RPN heads. Without extra training data or learnable components, we obtain significant boosts among strong baseline on COCO and LVIS datasets. Our method can be easily extended into other related domains, including open vocabulary instance/semantic segmentation. Rather than achieving STOA results, our goal is to fully explore the potential of VLM in the detector, which makes our method a generalized approach for various VLM and detectors.

## 5. Conclusion

This paper presents DST-Det, a novel open vocabulary object detection method incorporating a dynamic self-training strategy. By rethinking the pipeline of previous two-stage open vocabulary object detection methods, we identified a conceptual gap between training and testing for novel classes. To address this issue, we introduce a dynamic self-training strategy that generates pseudo labels for novel classes and iteratively optimizes the model with the newly discovered pseudo labels. Our experiments demonstrate that this approach can significantly improve the performance of mask average precision for novel classes, making it a promising approach for real-world applications. We hope our dynamic self-training strategy can help the community mitigate the gap between the training and testing for the OVOD task.

**Acknowledgement.** We thank the idea discussion of several friends, including Guangliang Cheng, Yining Li, and Kai Chen, in the earlier drafts. This work is supported by the National Key Research and Development Program of China (No.2023YFC3807603). This study is also supported

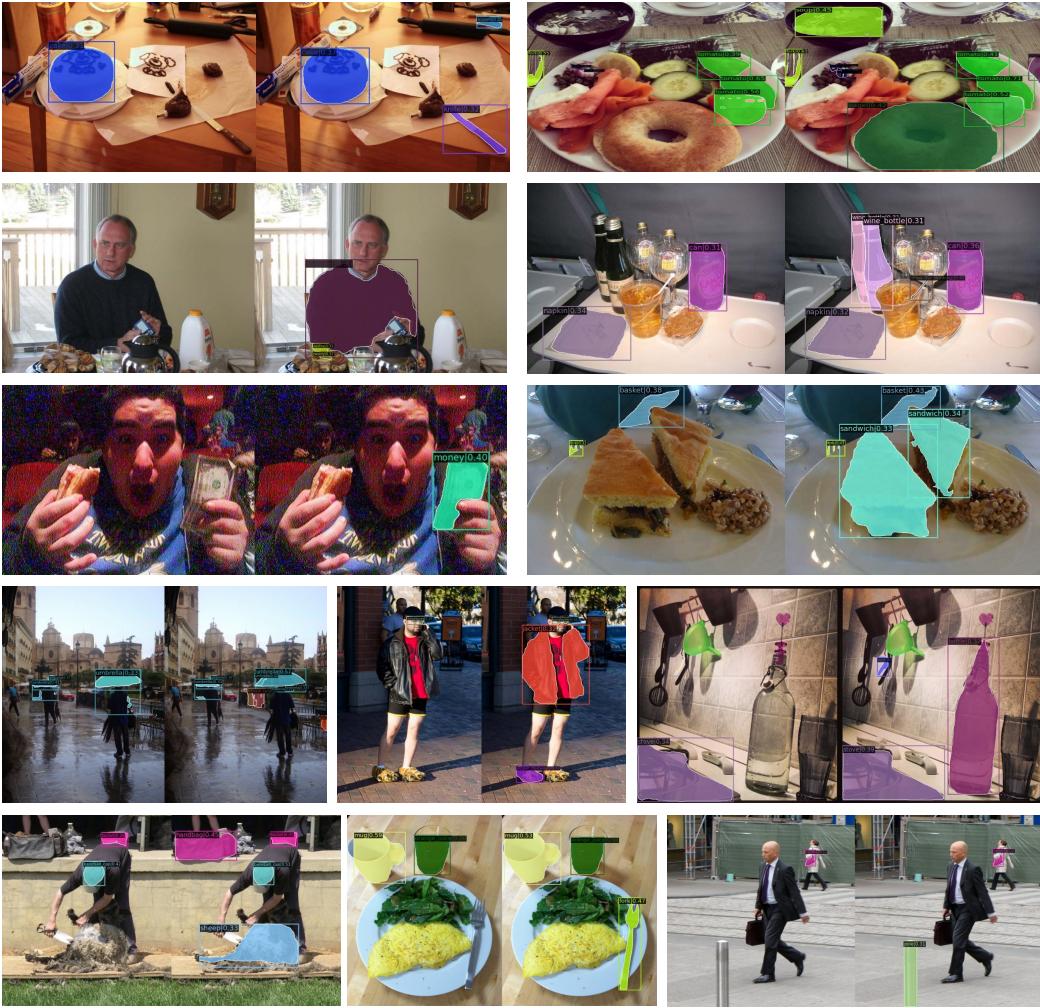


Figure 6. Improvement examples compared to baseline on LVIS dataset. The left is the baseline, and the right is our method.



Figure 7. Failure case visualization. The red boxes on the left images represent the ground truth annotations of novel classes; the right images are our predictions.

under the RIE2020 Industry Alignment Fund Industry Col-

laboration Projects (IAF-ICP) Funding Initiative, as well

as cash and in-kind contributions from the industry partner(s). It is also supported by Singapore MOE AcRF Tier 2 (MOE-T2EP20120-0001). We also gratefully acknowledge the support of SenseTime Research for providing the computing resources for this work.

## 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. In *NeurIPS*, 2022. 4
- [2] Relja Arandjelović, Alex Andonian, Arthur Mensch, Olivier J. H'enaff, Jean-Baptiste Alayrac, and Andrew Zisserman. Three ways to improve feature alignment for open vocabulary detection. *arXiv*: 2303.13518, 2023. 3
- [3] Maria Alejandra Bravo, Sudhanshu Mittal, and Thomas Brox. Localized vision-language matching for open-vocabulary object detection. *GCPR*, 2022. 3
- [4] Kyle Buettner and Adriana Kovashka. Enhancing the role of context in region-word alignment for object detection. *arXiv*: 2303.10093, 2023. 3
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3
- [6] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, 2020. 4
- [7] Peixian Chen, Kekai Sheng, Mengdan Zhang, Yunhang Shen, Ke Li, and Chunhua Shen. Open vocabulary object detection with proposal mining and prediction equalization. *arXiv*: 2206.11134, 2022. 3
- [8] Han-Cheol Cho, Won Young Jhoo, Woohyun Kang, and Byungseok Roh. Open-vocabulary object detection using pseudo caption labels. *arXiv*: 2303.13040, 2023. 3
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 6, 10
- [10] 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*, 2022. 3, 8
- [11] 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*, 2022. 3
- [12] 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*, 2022. 1, 2, 3, 4, 8, 9
- [13] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 4
- [14] 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, 6, 7, 8
- [15] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 3, 7, 10
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 3, 4, 5, 6, 7, 10
- [17] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *CVPR*, 2022. 4
- [18] 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. If you use this software, please cite it as below. 3
- [19] Chao Jia, Yinfen Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 3, 4
- [20] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Multi-modal classifiers for open-vocabulary object detection. *ICML*, 2023. 3
- [21] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Multi-modal classifiers for open-vocabulary object detection. In *ICML*, 2023. 8
- [22] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Contrastive feature masking open-vocabulary vision transformer. *ICCV*, 2023. 3
- [23] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, 2023. 3, 8
- [24] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 4
- [25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 4
- [26] Weicheng Kuo, Yin Cui, Xiuye Gu, A. J. Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023. 2, 3, 5, 7, 8
- [27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 4
- [28] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 4
- [29] Liunian Harold Li\*, Pengchuan Zhang\*, Haotian Zhang\*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 3

- [30] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *CVPR*, 2024. 3
- [31] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *CVPR*, 2023. 11
- [32] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, 2020. 3
- [33] 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. 2, 3, 8
- [34] Jiaying Lin and Shaogang Gong. Gridclip: One-stage object detection by grid-level clip representation learning. In *arXiv:2303.09252*, 2023. 3
- [35] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4, 7
- [36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 3
- [37] 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. 3
- [38] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, 2020. 3
- [39] Shilong Liu, Feng Li, Hao Zhang, Xiao Bin Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *ICLR*, 2022. 3
- [40] W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2015. 3
- [41] Zongyang Ma, Guan Luo, Jin Gao, Liang Li, Yuxin Chen, Shaoru Wang, Congxuan Zhang, and Weiming Hu. Open-vocabulary one-stage detection with hierarchical visual-language knowledge distillation. In *CVPR*, 2022. 3
- [42] Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *arXiv: 2306.09683*, 2023. 3
- [43] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022. 3
- [44] 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. In *ICML*, 2021. 1, 3, 4, 5, 10, 11
- [45] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022. 4
- [46] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NeurIPS*, 2022. 3, 8
- [47] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 3
- [48] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020. 3
- [49] 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
- [50] Cheng Shi and Sibei Yang. Edadet: Open-vocabulary object detection using early dense alignment. *ICCV*, 2023. 3, 8
- [51] Hengcan Shi, Munawar Hayat, and Jianfei Cai. Open-vocabulary object detection via scene graph discovery. *arXiv: 2307.03339*, 2023. 3
- [52] Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. Proposalsclip: unsupervised open-category object proposal generation via exploiting clip cues. In *CVPR*, 2022. 4
- [53] Hwanjun Song and Jihwan Bang. Prompt-guided transformers for end-to-end open-vocabulary object detection. *arXiv: 2303.14386*, 2023. 3
- [54] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. SparseR-CNN: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 3
- [55] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *CVPR*, 2021. 3
- [56] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, 2020. 3
- [57] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020. 3
- [58] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. In *TPAMI*, 2021. 3
- [59] Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. *ICCV*, 2023. 2, 3, 6, 7
- [60] 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*, 2023. 8
- [61] Tao Wang and Nan Li. Learning to detect and segment for open vocabulary object detection. *CVPR*, 2023. 3
- [62] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, 2020. 3

- [63] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jin-qiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. In *CVPR*, 2021. 3
- [64] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey. *TPAMI*, 2024. 1
- [65] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *ACM-MM*, 2020. 3
- [66] 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, 3, 8
- [67] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *arXiv preprint arXiv:2311.03352*, 2023. 3, 8
- [68] 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. 2, 3, 8
- [69] Jiahao Xie, Wei Li, Xiangtai Li, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Mosaicfusion: Diffusion models as data augmenters for large vocabulary instance segmentation. *arXiv preprint arXiv:2309.13042*, 2023. 4
- [70] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models. In *CVPR*, 2023. 2, 4
- [71] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv: 2308.02487*, 2023. 2, 3
- [72] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, 2022. 3, 4, 8
- [73] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 1, 3, 6, 7, 8
- [74] Hao Zhang, Feng Li, Xueyan Zou, Siyi Liu, Chun yue Li, Jianfeng Gao, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. *rXiv: 2303.08131*, 2023. 3
- [75] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2019. 3
- [76] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021. 3
- [77] Shiyu Zhao, Samuel Schulter, Long Zhao, Zhixing Zhang, Yumin Suh, Manmohan Chandraker, Dimitris N Metaxas, et al. Improving pseudo labels for open-vocabulary object detection. *arXiv preprint arXiv:2308.06412*, 2023. 1, 3, 4
- [78] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, Vijay Kumar B.G, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N. Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, 2022. 1, 2, 3, 8, 9
- [79] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luwei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 3, 8
- [80] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 4
- [81] Hao Zhou, Tiancheng Shen, Xu Yang, Hai Huang, Xiangtai Li, Lu Qi, and Ming-Hsuan Yang. Rethinking evaluation metrics of open-vocabulary segmentaion. *arXiv preprint arXiv:2311.03352*, 2023. 3
- [82] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 1, 3, 4, 6, 8
- [83] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. In *arXiv:2103.07461*, 2021. 3
- [84] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv:1904.07850*, 2019. 3
- [85] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 3
- [86] Yi Zhu, Karan Sapra, Fitzsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 4