Learning to Prompt for Open-Vocabulary Object Detection with Vision-Language Model

Yu Du[1] Fangyun Wei[2†] Zihe Zhang[1] Miaojing Shi[3†] Yue Gao[2] Guoqi Li[1]
[1]Tsinghua University [2]Microsoft Research Asia [3]King's College London
{duyu20, zh-zhang17}@mails.tsinghua.edu.cn liguoqi@mail.tsinghua.edu.cn
{fawe, yuegao}@microsoft.com miaojing.shi@kcl.ac.uk

Abstract

Recently, vision-language pre-training shows great potential in open-vocabulary object detection, where detectors trained on base classes are devised for detecting new classes. The class text embedding is firstly generated by feeding prompts to the text encoder of a pre-trained vision-language model. It is then used as the region classifier to supervise the training of a detector. The key element that leads to the success of this model is the proper prompt, which requires careful words tuning and ingenious design. To avoid laborious prompt engineering, there are some prompt representation learning methods being proposed for the image classification task, which however can only be sub-optimal solutions when applied to the detection task. In this paper, we introduce a novel method, detection prompt (DetPro), to learn continuous prompt representations for open-vocabulary object detection based on the pre-trained vision-language model. Different from the previous classification-oriented methods, DetPro has two highlights: 1) a background interpretation scheme to include the proposals in image background into the prompt training; 2) a context grading scheme to separate proposals in image foreground for tailored prompt training. We assemble DetPro with ViLD, a recent state-of-the-art open-world object detector, and conduct experiments on the LVIS as well as transfer learning on the Pascal VOC, COCO, Objects365 datasets. Experimental results show that our DetPro outperforms the baseline ViLD [7] in all settings, e.g., +3.4 AP^{box} and +3.0 AP^{mask} improvements on the novel classes of LVIS. Code and models are available at https://github.com/dyabel/detpro.

1. Introduction

Object detection aims at locating bounding boxes of objects in an image as well as assigning labels to them. In last few years, object detection [19, 20] achieves great success in solving the closed-set problem, *i.e.*, detectors can detect classes present in the training set. To increase the detection vocabulary, the common practice is by collecting more data with desired classes. Besides the expensive labeling cost in this process, it often leads to a long-tailed distribution [8, 13] of object classes: detectors need to be carefully designed to avoid overfitting on frequently-occurred categories in the dataset. In contrast, an alternative way for increasing the detection vocabulary is open-vocabulary object detection (OVOD), where detectors are trained on base classes and equipped with ability to detect new classes.

Recently, ViLD [7] introduces a framework for open-vocabulary object detection, which distills the knowledge from a pre-trained vision-language model into a detector. It is inspired by the recent progress of vision-language pre-training, e.g, CLIP [18] and ALIGN [10], where two separate encoders, namely image encoder and text encoder, are used to maximize the alignment between images and corresponding texts. In ViLD's implementation, they feed text descriptions of base classes, known as prompt, into the text encoder of CLIP to generate the class text embedding. The embedding is then utilized to classify object proposals and supervise the detector training. To perform open-set object detection, the base class text embedding is replaced with the embedding of both base and novel classes. The prompt design, also known as prompt engineering, is crucial in this process as we observe a slight word change in it would end up with clear positive or negative impact on the detection performance. Designing proper prompts requires domain expertise and carefully word tuning from human, as of [7]. To avoid such high-end and rather laborious demand from human, the alternative way is to automatically learn prompt's context using continuous representations, we name it as prompt representation learning in our work.

In this paper, we propose a novel method named detection prompt (DetPro) to learn prompt representations, in the setting of open-vocabulary object detection with pre-trained

†Corresponding author.

vision-language model (OVOD-VLM). There are some recent works focusing on prompt representation learning such as CoOp [38], who targets for improving image classification accuracy based on the pre-trained vision-language models. Directly applying CoOP into the OVOD-VLM is not realistic: image classification only needs to recognize the correct labels of input images while object detection requires detectors to distinguish foregrounds from backgrounds, and classify region proposals in foregrounds into different object classes. We thus introduce a new Detection Prompt (DetPro) to automatically learn prompt representations in OVOD-LVM based on positive and negative proposals w.r.t. ground truth in images.

Prompt learning in object detection faces two critical issues: 1) Negative proposals, despite being very important to object detection, do not correspond to specific object classes, therefore can not be easily included into the prompt learning process. 2) Unlike objects in image classification being centered and big in images, objects in positive proposals are often associated with different levels of contexts, learning one prompt context for these proposals can not be sufficient. To tackle them, we introduce,

- a background interpretation scheme for negative proposal inclusion, which optimizes the embedding of negative proposals to be away from all other class embedding;
- a context grading scheme with tailored positive proposals, which tailors the prompt representation learning with different positive proposal sets corresponding to different context levels.

We assemble DetPro with ViLD [7], and conduct a series of experiments on LVIS and transfer the LVIS-trained model to other datasets including Pascal VOC, COCO and Objects365. In all settings, our DetPro outperforms the ViLD, e.g., +3.4 AP^{box} and +3.0 AP^{mask} improvements on the novel classes of LVIS.

2. Related Work

Prompt Learning. Recently, the development of large vision-language model (VLM), *e.g.*, CLIP [18] and ALIGN [10], emerges and finds its applications in few-shot or zero-shot learning tasks [5, 28]. The VLMs are trained on huge amount of image-text pairs collected from web and contrastive learning [11] is adopted to align the image and text embedding. The pretrained VLMs can be transferred to its downstream tasks with either finetuning [16, 26] or prompt engineering [38]. A task-specific prompt can boost the performance significantly [18] but requires laborious prompt engineering. Inspired by prompt learning in language tasks, CoOp [38] proposes the context optimization to automate prompt engineering for few-shot classification. It models the context of prompts as continuous representations that are end-to-end learned from a small set of data. This paper extends CoOP to OVOD by designing special strategies to handle foreground and background proposals within images. While CoOP learns the prompt with samples of all categories our DetPro is trained on only base classes and expected to generalize to novel classes.

Open-Vocabulary Object Detection. Despite the remarkable success of DNNs [2, 9, 12, 24] in the computer vision field, they often require a large amount of annotated data in order to get satisfying results of object detection [3, 14, 20, 32]. To alleviate the reliability of DNNs on big data and elaborate annotations, different paradigms such as semi-supervised learning [34], few-shot learning [25, 27, 36], zero-shot learning [21,30,33], self-supervised learning [31], open-set learning [6, 22, 29] and advanced training strategies [17, 35] are introduced.

Particularly, for the zero-shot detection task, it aims to generalize from seen classes (with bounding box annotations) to unseen classes. Despite they have made some progress, their overall performance is still far behind the fully-supervised methods [1, 39], therefore research on it is not flourishing yet.

Recently, open-vocabulary object detection emerges as a more general and practical paradigm onto the stage than the zero-shot detection: an unbounded vocabulary of concepts is firstly acquired by training on image-text pairs, then the detector is required to detect novel classes with the availability of bounding box annotations of a number of base classes. The representative solutions include OVR-CNN [37] and ViLD [7]. OVR-CNN [37] pretrains the backbone using a corpus of image-caption pairs and finetunes the detector with annotations of only a few object categories while ViLD [7] directly distills knowledge from a pretrained open-vocabulary classification model into a two-stage detector.

We place our work in the OVOD setting and build our solution upon the ViLD [7]. ViLD uses hand-crafted prompts for generating class embedding, while we design fine-grained automatic prompt learning and special background interpretation to find the desired prompts.

3. Problem Setting

The goal of DetPro is to learn continuous prompt representations for OVOD-VLM. Figure 1 shows an overview of our DetPro, which includes two key elements: background interpretation for negative proposal inclusion, foreground context grading with tailored positive proposals. They are dedicated to the positive and negative losses in the figure. Afterwards, we devise DetPro upon the recent OVOD pipeline ViLD [7] in Figure 2 where DetPro serves as a replacement for the proposal classifier in ViLD to realize automatic prompt engineering for it.

Data Split. We divide categories in a detection dataset into two disjoint sets for base classes $\mathcal{C}_B$ and novel classes $\mathcal{C}_N$.

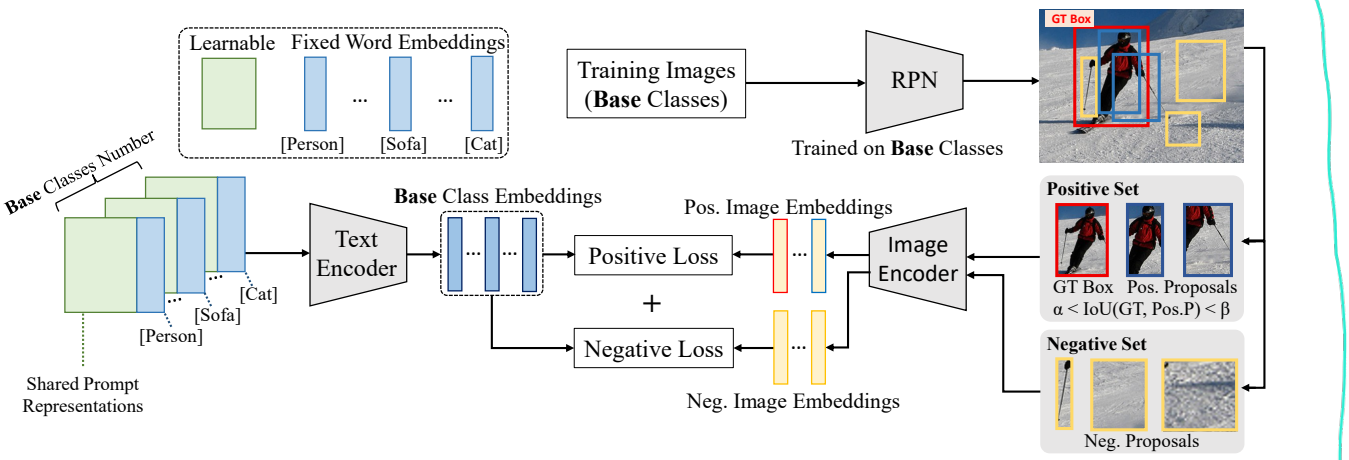


Figure 1. Overview of DetPro. Positive loss is defined between visual embedding of positive proposals in images and their corresponding class embedding; while negative loss is defined between visual embedding of negative proposals and all class embedding. Different tailored positive proposal sets ($\alpha <$ IoU(GT, Pos P) $< \beta$) are used to learn different prompt representations and are ensemabled in the end.

We use $|\mathcal{C}_B|$ and $|\mathcal{C}_N|$ to denote the number of base and novel classes, respectively. Correspondingly, we have $\mathcal{X}_T$ and $\mathcal{X}_I$ for the training and inference dataset, respectively. $\mathcal{X}_T$ contains only base classes $\mathcal{C}_B$ with annotations for training, while $\mathcal{X}_I$ contains both $\mathcal{C}_B$ and $\mathcal{C}_N$ for the trained model to recognize objects from both $\mathcal{C}_B$ and $\mathcal{C}_N$.

Pre-trained Vison-language Model. We use CLIP [18] as our vision-language model, which consists of a text encoder $\mathcal{T}(\cdot)$ and an image encoder $\mathcal{I}(\cdot)$. $\mathcal{T}(\cdot)$ takes the input of prompt representation of a class and outputs the corresponding text embedding, which is also named as class embedding in our work; $\mathcal{I}(\cdot)$ takes the input of an image of size 224×224 and outputs the corresponding image embedding.

Detection Framework. We adopt Faster-RCNN with ResNet-50 and FPN as our detector.

4. Method

We first review prompt representation learning in image classification, then we present our DetPro in object detection; finally, assemble it onto ViLD for OVOD.

4.1. Preliminaries: Prompt

Original CLIP [18] feeds human-defined prompt, *e.g.* 'a photo of [CLASS]', into its text encoder $\mathcal{T}$ to generate the class embedding for image classification. In a specific case, [CLASS] is replaced by the class name such as 'person' and 'cat'. Identifying the proper prompt is a non-trivial task, which often costs a significant amount of time for words tuning. To bypass it, CoOp [38] proposes to automatically learn prompt representations. The learnable prompt representation $\boldsymbol{V}_c$ for given class $c \in C_B$ is defined as follows:

$$\boldsymbol{V}_c = [\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_L, \boldsymbol{w}_c], \tag{1}$$

where $\boldsymbol{v}_i$ denotes the i-th leanable context vector, $\boldsymbol{w}_c$ the fixed class token of base class c and L the context length. $[\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_L]$ can be analogue to context of the human-defined prompt, *e.g.* 'a photo of', while $\boldsymbol{w}_c$ analog to the class name [CLASS]. $\{\boldsymbol{v}_i\}_{i=1}^{L}$ are randomly initialized to have the same dimension to the word embedding $\boldsymbol{w}_c$ (512 in this work). The learned prompt context $[\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_L]$ is shared across classes, such that when a new class comes, its prompt representation can be easily obtained by (1). The class embedding $\boldsymbol{t}_c$ of class c is generated by feeding $\boldsymbol{V}_c$ into the CLIP text encoder $\mathcal{T}(\cdot)$:

$$\boldsymbol{t}_c = \mathcal{T}(\boldsymbol{V}_c). \tag{2}$$

In image classification task, given an image x, we can first feed it into the CLIP image encoder $\mathcal{I}(\cdot)$ to extract its image embedding $\boldsymbol{f}$. Assuming this image belongs to class c, the probability of $\boldsymbol{f}$ being classified as class c is computed as:

$$p_c = \frac{\exp(\cos(\boldsymbol{f}, \boldsymbol{t}_c)/\tau)}{\sum_{i \in \mathcal{C}_B} \exp(\cos(\boldsymbol{f}, \boldsymbol{t}_i)/\tau)}, \tag{3}$$

where τ is a temperature parameter, $\cos(\cdot, \cdot)$ denotes the cosine similarity. The cross entropy loss is applied to optimize $[\boldsymbol{v}_1, \boldsymbol{v}_2 \dots \boldsymbol{v}_L]$ while both $\mathcal{I}(\cdot)$ and $\mathcal{T}(\cdot)$ are fixed:

$$\mathcal{L}_p = -\log p_c. \tag{4}$$

4.2. Detection Prompt

Naïve Solution. Object detection differs from image classification as for each training image we have class labels provided on ground truth bounding boxes of objects, and for each test image we need to localize bounding boxes of objects and predict class labels for them. To adapt the prompt representation learning strategy CoOP [38] into the detection task, the straightforward way is to simulate the classification scenario that it works: given an image x, we instead feed its cropped ground truth bounding boxes into the CLIP

image encoder $\mathcal{I}(\cdot)$ to obtain the box embedding $\boldsymbol{f}$, respectively. Each ground truth box belongs to only one object class c; we denote by $\mathcal{G}$ all ground truth bounding boxes over images. We can then follow the same equations (3,4) to learn a region-level classifier on $\mathcal{G}$. This classifier can be further assembled with an established object detection pipeline (*e.g.* Faster R-CNN), specified in Section 4.3.

This naïve adaption can work to certain extent, but is only a sub-optimal solution: the rich information in images apart from the ground truth bounding boxes has been dropped, including foreground and background proposals, this however is essential to learn a robust region-level (proposal) classifier for detection.

Fine-grained Solution. In order to make use of image proposals, we first train a RPN on base classes $\mathcal{C}_B$ to extract them from $\mathcal{X}_T$. Foreground proposals $\mathcal{F}$ are those whose IoUs w.r.t. one ground truth in $\mathcal{G}$ are larger than a thresh, *i.e.* 0.5, while background proposals $\mathcal{B}$ are negative proposals whose IoUs w.r.t. all ground truth in $\mathcal{G}$ are smaller than the thresh. We call the union of $\mathcal{F}$ and $\mathcal{G}$ the positive proposal set $\mathcal{P}$, *i.e.* $\mathcal{P} = \mathcal{F} \cup \mathcal{G}$, and $\mathcal{B}$ the negative proposal set $\mathcal{N}$, *i.e.* $\mathcal{N} = \mathcal{B}$. For a proposal in $\mathcal{P}$, unless it's the ground truth whose target object inside is tightly bounded, it normally includes a big partial of the object with considerable surrounding context. The positive proposals thus vary a lot in contexts depending on their IoUs w.r.t. the ground truth. This shall result into different visual embedding when feeding them into $\mathcal{I}(\cdot)$. Consequently, different prompt representations should also be learned dedicated to different prompt contexts. To address this issue, we introduce a context grading scheme with tailored positive proposals (specified later). On the other hand, for a proposal in $\mathcal{N}$, it contains mostly background with the possibility of a small partial of target objects. The background does not have a specific class name, thus its prompt representation can not be directly obtained (no $\boldsymbol{w}_c$ in Eq.1), nor does its class embedding. Negative proposals serve as a very important role in object detection. In order to utilize them in our detection prompt, we introduce a background interpretation scheme for negative proposal inclusion. Below we detail it.

Background interpretation for negative proposal inclusion. Background might contain some object classes inside, but they can not be normally recognized as consequences of being either too small, too incomplete or too vague. In other words, given an negative proposal n, its image embedding $\boldsymbol{f}_n$ by $\mathcal{I}(\cdot)$ should be dissimilar to any text embedding $\boldsymbol{t}_c$ of other classes by $\mathcal{T}(\cdot)$.

The probability p_{nc} of $\boldsymbol{f}_n$ being classified as class c is computed via Eq.3. We want any p_{nc} to be small; in practice, since $|\mathcal{C}_B|$ is big, we could simply optimize any p_{nc} to $\frac{1}{|\mathcal{C}_B|}$. This forces the negative proposal to be equally unlike any object classes. The loss function is thus formulated as,

$$\mathcal{L}_{\text{n}} = -\sum_{c=1}^{|\mathcal{C}_B|} w \log p_{nc}, \quad w = \frac{1}{|\mathcal{C}_B|}. \tag{5}$$

An alternative way for background interpretation is to learn a stand alone background prompt representation $\boldsymbol{V}_{\text{bg}}$ which is similar to $\boldsymbol{V}_c$ for class c but without the class token:

$$\boldsymbol{V}_{\text{bg}} = [\boldsymbol{v}_1^{\text{bg}}, \boldsymbol{v}_2^{\text{bg}}, \dots, \boldsymbol{v}_L^{\text{bg}}]. \tag{6}$$

Similarly, we use Eq.2 to generate background embedding $\boldsymbol{t}_{\text{bg}}$ and feed the negative proposal n into $\mathcal{I}(\cdot)$ to generate $\boldsymbol{f}_n$. The probability p_{nbg} is computed as:

$$p_{\text{nbg}} = \frac{\exp\left(\cos(\boldsymbol{f_n}, \boldsymbol{t}_{\text{bg}})/\tau\right)}{\sum_{c=1}^{\mathcal{C}_B} \exp\left(\cos(\boldsymbol{f_n}, \boldsymbol{t}_c)/\tau\right) + \exp\left(\cos(\boldsymbol{f_n}, \boldsymbol{t}_{bg})/\tau\right)}. \tag{7}$$

The negative loss is defined as:

$$\mathcal{L}_{\text{n}} = -\log p_{\text{nbg}}. \tag{8}$$

This alternative way is inferior to the first way. The background content may vary a lot, the second way learns an explicit background embedding to let all negative proposals be close to it, which can not be sufficient. In contrast, in the first way it is implicitly interpreted to let each negative proposal being away from all other class embedding, which can be more robust.

Context grading with tailored positive proposals. A positive proposal may incorporate different contexts w.r.t. the target object. This difference can be analogue in the prompt context: given a ground truth bounding box of an object class, we can say 'a photo of [CLASS]'; while given a foreground proposal of a partial object, we could instead say 'a photo of partial [CLASS]'. The learned prompt context representations for 'a photo of' and 'a photo of partial' will be different, which ends up with different class embedding for the two types of prompts. They should be optimized with positive proposals corresponding to different levels of contexts, respectively. We introduce a foreground context grading scheme with tailored positive proposals for this purpose.

Specifically, we divide the positive proposals of the IoU range $[a, b]$ into K disjoint groups with an IoU interval of t, such that $K = (a - b)/t$. The foreground context will be graded in different groups, such that positive proposals within each group have similar context level w.r.t. their respective ground truth. We therefore learn prompt representations in the K groups independently. Within the k-th group, we follow the same equations (3,4) to extract visual embedding $\boldsymbol{f}_p$, compute probability p_{pc}, and optimize positive loss $\mathcal{L}_p$ for any positive proposal p of class c inside. The same negative proposal set $\mathcal{N}$ is included into each group

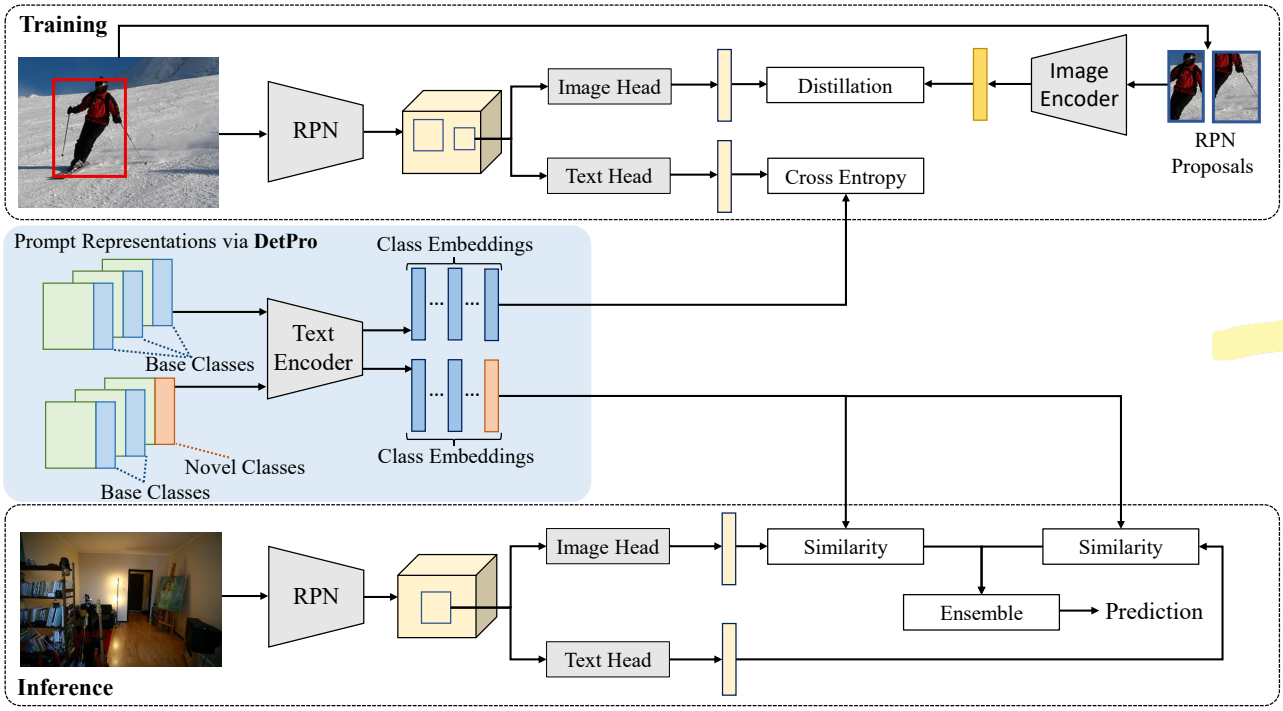


Figure 2. Assembling DetPro with ViLD. DetPro is highlighted with azure background. We omit the class-agnostic bounding box regression branch and mask prediction branch in both training and testing pipelines.

such that the final loss function within each group is,

$$\mathcal{L} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \mathcal{L}_n + \frac{1}{|\mathcal{P}^k|} \sum_{p \in \mathcal{P}^k} \mathcal{L}_p. \tag{9}$$

The prompt representation $\boldsymbol{V}_c^k$ is learned in each group for class c. In the end, the learned representations are ensembled over K groups by average, such that $\boldsymbol{V}_c = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{V}_c^k$.

4.3. Assembling DetPro onto ViLD

ViLD [7] is a recent framework for OVOD. It distills the knowledge from CLIP [18] into a two stage detector, *i.e.*, Faster R-CNN [20]. Figure 2 shows assembling our DetPro with ViLD.

Training ViLD with DetPro. A learned DetPro generates prompt representations based on Eq. 1 for base classes, which we can feed into $\mathcal{T}(\cdot)$ to generate base class embedding. The embedding is used as proposal classifier for the detector. Following ViLD, we employ two R-CNN heads (sub-branches), namely image head and text head. The image head distills knowledge from CLIP image encoder, while the text head replaces the original R-CNN classifier by our base class embedding (fixed) plus a learnable background embedding (see Figure 2).

We briefly describe the training process as in [7]: for each region proposal generated by RPN, we pass it through the text head and image head respectively to extract two RoI features for subsequent loss computations. There are two losses: for the text head, cosine similarities between RoI features and base class embedding is computed for classification and a standard cross entropy loss $\mathcal{L}_{\text{text}}$ is adopted. As for the image head branch, we crop and resize proposals generated by the RPN, and feed them into $\mathcal{I}(\cdot)$ to generate image embedding. A L1 loss (*i.e.* $\mathcal{L}_{\text{image}}$) is applied to minimize the distance between image embedding and the corresponding RoI feature extracted by image head. The generation of image embedding can be performed offline by using a pre-trained RPN. The overall classification loss is the weighted sum of $\mathcal{L}_{\text{text}}$ and $\mathcal{L}_{\text{image}}$. In addition, we replace the second-stage class-specific bounding box regression and mask prediction layers with class-agnostic modules. A standard regression loss and mask prediction loss are also utilized during training.

Inference ViLD with DetPro. At inference stage, we use Eq. 6 to generate prompt representations for both base and novel classes, and class embedding is extracted by feeding prompt representations into $\mathcal{T}(\cdot)$. Thanks to the shared context vectors, prompt representations optimized by DetPro can be well generalized to novel classes though trained on only base classes. Given a test image x, RPN first generates a set of proposals. We pass each proposal through the text head and the image head to extract two RoI features (see Figure 2). For each one, we compute its cosine similarities to all class embedding to obtain confidence scores. The final probability for x is the geometric mean of two confidence

Method	Epoch	Detection				Instance segmentation			
		AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f	AP
Supervised (base)	20	0.0	26.1	34.0	24.7	0.0	24.7	29.8	22.4
Supervised (base+novel)	20	15.5	25.5	33.6	27.0	16.4	24.6	30.6	25.5
ViLD (base) [7]	460	16.7	26.5	34.2	27.8	16.6	24.6	30.3	25.5
ViLD* (base) [7]	20	17.4	27.5	31.9	27.5	16.8	25.6	28.5	25.2
DetPro (base)	20	**20.8**	27.8	32.4	28.4	**19.8**	25.6	28.9	25.9

Table 1. Comparison with ViLD on LVIS v1 dataset. * denotes our re-implementation version, see Section 5.2 for the details. The frequent and common classes are used as the base classes, while the rare classes are held out as the novel classes. AP_r is the main evaluation metric for open-world object detection.

Method	Pascal VOC		COCO						Objects365					
	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Supervised	78.5	49.0	46.5	67.6	50.9	27.1	67.6	77.7	25.6	38.6	28.0	16.0	28.1	36.7
ViLD* [7]	73.9	**57.9**	34.1	52.3	36.5	21.6	38.9	46.1	11.5	17.8	12.3	4.2	11.1	17.8
DetPro	**74.6**	**57.9**	**34.9**	**53.8**	**37.4**	**22.5**	**39.6**	**46.3**	**12.1**	**18.8**	**12.9**	**4.5**	**11.5**	**18.6**

Table 2. We evaluate the LVIS-trained model on Pascal VOC test set, COCO validation set and Object365 validation set.

scores.

5. Experiment

5.1. Dataset and Evaluation Metrics

We conduct our main experiments on LVIS v1 [8] dataset. DetPro and its open-vocabulary object detector are trained on LVIS base classes. We evaluate our approach on LVIS novel classes. Meanwhile, we conduct transfer experiments to demonstrate generalization ability of our approach, and evaluate our LVIS-trained model on Pascal VOC [4] test set, COCO [15] validation set and Objects365 [23] validation set.

LVIS V1 Dataset. LVIS v1 is a large-scale object detection dataset with a long-tail data distribution. It divides the categories into 'frequent', 'common', 'rare' according to their appearing frequency in the training set. Following ViLD [7], The frequent and common classes are used as the base classes (866 classes), while the rare classes are held out as the novel classes (337 classes).

Pascal VOC Dataset. Pascal VOC is a collection of datasets (including VOC2007 and VOC2012) for object detection which contains 20 object categories.

COCO. COCO is a standard dataset comprising 80 categories of common objects in natural context. It contains ∼118k images with bounding box and instance segmentation annotations. Following ViLD [7], the instance masks are not computed.

Objects365 Dataset. Objects365 is a brand-new large-scale object detection dataset with 365 categories and high-quality bounding box annotations.

Evaluation Metrics. We use average precision (AP) to evaluate the performance of object detection and instance segmentation. For LVIS experiments, AP_r is the main indicator, we report the results of AP_f and AP_c as well. While for transfer experiments on Pascal VOC, COCO and Objects365, we use AP, AP_{50}, AP_{75}, AP_s, AP_m and AP_l as the evaluation metrics.

5.2. Implementation Details

DetPro. Unless otherwise specific, we use the following settings of DetPro: context length of 8; class token in the end; 10% of the background proposals; implicitly model background by Eq. 5. The context vectors are initialized by drawing from a zero-mean Gaussian distribution of standard deviation 0.02. We choose SGD optimizer with an initial learning rate of 0.002 which is decayed by the cosine annealing rule. We train our DetPro for 6 epochs.

ViLD and Object Detector. We use the Mask R-CNN with ResNet-50 and FPN as our detector. The model is trained on 8 GPUs with 2 images per GPU. Synchronized batch normalization is used. We use SGD as the optimizer, the momentum and the weight decay are set to 0.9 and 0.00003, respectively. For the comparison with state-of-the-art methods, our detector is trained for 20 epochs, and we train 12 epochs for ablation studies. The learning rate is initialized as 0.02, it is divided by 10 at 16-th epoch and 8-th epoch for the 20-epoch and 12-epoch schedule, respectively. A warm up step with learning rate of 0.001 is performed for the first 500 iterations. We re-implement the ViLD [7], named ViLD*, by replacing the pre-trained ResNet-50 with self-supervised pre-trained SoCo [31] to reduce the huge training cost. In the original implementation of ViLD, the whole training process takes up to 180,000 iterations with batchsize of 256, approximately 460 epochs, which is unaffordable. In our re-implementation, the training epoch is reduced from 460 to 20 while comparable performance is achieved.

Strategy	AP_r	AP_c	AP_f	AP
DetPro w/o BG	16.9	25.1	27.7	24.7
DetPro-LearnableBG	15.3	**25.4**	27.9	24.6
DetPro-SoftBG	**19.1**	**25.4**	**28.2**	**25.4**

Table 3. Ablation study on different strategies of involving negative proposals into our DetPro.

Background proposals	AP_r	AP_c	AP_f	AP
10%	**19.1**	25.4	28.2	**25.4**
30%	18.3	**25.6**	**28.4**	**25.4**
50%	17.8	**25.6**	**28.4**	**25.4**
100%	17.6	25.1	28.2	25.0

Table 4. Ablation on number of background proposals involved in DetPro training.

GT	FG	BG	AP_r	AP_c	AP_f	AP
✓			15.3	**25.4**	27.9	24.6
✓	✓		16.9	25.1	27.7	24.7
✓		✓	17.7	25.3	**28.2**	25.1
✓	✓	✓	**19.1**	**25.4**	**28.2**	**25.4**

Table 5. Ablation study on the involvement of different training data. 'GT': ground-truth; 'FG': foreground; 'BG': background.

Vision-Language Model. We use the publicly available CLIP[1] as pre-trained vision-language model. We adopt the ViT-B/32 as the image encoder.

5.3. Main Results

Experiment on LVIS v1 Dataset. Table 1 shows the comparison with ViLD on the LVIS v1 dataset. Our re-implementation version of ViLD (denoted as ViLD *) achieves comparable AP compared with the original implementation, while reducing the training epochs from 460 to 20. Note our performance on AP_r is even slightly higher while the high AP_c and AP_f of original ViLD is owed to the large scale jittering augmentation with long training schedule (approximately 460 epochs). Our DetPro improves the baseline ViLD* by +3.4 AP_r on object detection, and +3.0 AP_r on instance segmentation, respectively.

Transfer to Other Datasets. Following ViLD [7], we conduct experiments on transferring LVIS-trained DetPro to other datasets, namely Pascal VOC 2007 test set, COCO validation set and Objects365 v1 validation set, by directly replacing the class tokens. As is shown in Table 2, Our DetPro improves the baseline ViLD* on all three datasets on Pascal VOC, COCO and Objects365, demonstrating the effectiveness and generalization of our DetPro.

5.4. Ablation Study

We use LVIS setting, where our model is trained on the LVIS base classes and evaluated on the LVIS rare classes, for all ablation studies. We report the results of instance segmentation. AP_r is used as the main indicator to evaluate the generalization of DetPro.

Different Ways for Background Interpretation. As described in Section 4.2, we introduce two strategies to include negative (background) proposals, namely DetPro-SoftBG (Eq. 5) and DetPro-LeanableBG (Eq. 7,8). Table 3 compares two variants with a baseline named DetPro w/o BG, in which neither negative set nor negative loss are used. DetPro-SoftBG ourperforms the baseline by +2.2 AP_r, which demonstrates the importance of involving background in a proper way. We observe that DetPro-LearnableBG is worse than the baseline by -1.6 AP_r. We conjecture that background content may vary a lot, learning an explicit background embedding to let all negative proposals be close to it, which can not be sufficient.

[1]https://github.com/openai/CLIP

Number of Negative Proposals. We already demonstrate the importance of involving negative proposals into our DetPro, but how many of them should we use in training? Table 4 shows the study. The result on AP_r consistently declines with the increasing of negative samples. Since negative samples are significantly more than positive ones, reducing negatives can avoid bias towards background as well as speed up training. Our default is 10.

Involvement of Different Training Data. We study various combinations of training data in Table 5. Our default setting, *i.e.*, including ground-truth, foreground proposals and background proposals, yields the best performance among others. Eliminating either foreground proposals or background proposals from training data leads to performance degradation. Using only ground-truth for training degenerates to CoOp [38].

Context Grading and Prompt Representation Ensemble. Here we study the effects of prompt representation ensemble as shown in Table 6. As described in Section 4.2, we divide the positive proposals of the IoU range $[a, b]$ into K disjoint groups with an IoU interval of t. Then class embedding from K learned DetPro are ensembled. From the table we observe that our DetPro with ensemble strategy consistently improve the performance over their non-ensemble counterparts, *e.g.* 'Ensemble (0.5:1.0:0.1)' outperforms 'IoU range = [0.5-1.0]' by +3.0 AP_r. The main improvements come from the novel classes.

Context Length. We study the effects of using different context lengths L. We vary the length from 4 to 8 to 16 and Table 7 shows the study. CoOp [38] has shown that using longer prompt can lead to better performance on close-vocabulary image classification task. We obtain the same conclusion from the performance of base classes (AP_c and AP_f). However, it does not hold true for novel classes, suggesting that longer prompt may cause over-fitting to base categories. We set context length as 8 by default.

Position of Class Token. Table 8 studies inserting class

IoU range	AP_r	AP_c	AP_f	AP
0.5-0.6	17.3	25.3	28.2	25.0
0.6-0.7	18.0	25.4	28.1	25.4
0.7-0.8	17.2	25.4	28.3	25.1
0.8-0.9	17.3	24.9	28.2	24.9
0.9-1.0	17.2	25.2	28.3	25.0
0.5-1.0	16.1	25.7	28.3	25.1
0.6-1.0	17.2	25.4	28.9	25.3
0.7-1.0	16.8	25.0	28.3	25.1
0.8-1.0	17.2	25.2	28.4	25.1
Ensemble (0.5:1.0:0.1)	**19.1**	25.4	28.2	25.4
Ensemble (0.6:1.0:0.1)	18.4	25.2	28.2	25.2
Ensemble (0.7:1.0:0.1)	18.7	**25.8**	**28.3**	**25.5**
Ensemble (0.8:1.0:0.1)	18.2	25.3	28.1	25.2

Table 6. The effects of prompt representation ensemble. 'Ensemble (0.5:1.0:0.1)' represents we divide the positive proposals of the IoU range [0.5-1.0] into 5 disjoint groups with an IoU interval of 0.1. Then we use each group to train a separate DetPro and perform ensemble on 5 learned models.

Length	AP_r	AP_c	AP_f	AP
4	18.7	24.9	28.2	25.1
8	**19.1**	**25.6**	**28.3**	25.2
16	17.7	**25.6**	**28.3**	**25.3**

Table 7. Ablation study on context lengths.

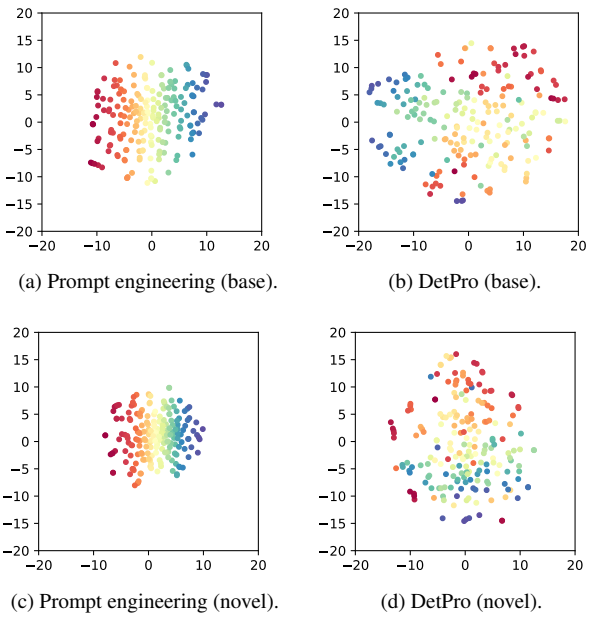
Position	AP_r	AP_c	AP_f	AP
Front	16.4	24.5	**28.3**	24.6
Middle	18.0	25.1	**28.3**	25.1
End	**19.1**	**25.4**	28.2	**25.4**

Table 8. Ablation study of inserting class token into different positions of prompt representation.

token into different positions, namely front, middle and end, of the prompt representations. Generally, the best position depends on the dataset [38]. In our experiment, positioning class token in the end achieves the best performance.

5.5. Visualization

To further demonstrate the importance of involving both foreground and background proposals in detection-oriented prompt representation learning. We randomly select 200 base classes and 200 novel classes from LVIS dataset and use t-SNE to visualize the class embedding generated by DetPro and prompt engineering as shown in Figure 3. We observe that the class embedding generated by DetPro is more discriminative in the embedding space, this superior property indicates they are more capable of being region classifiers for open-vocabulary object detector.



(a) Prompt engineering (base). (b) DetPro (base).

(c) Prompt engineering (novel). (d) DetPro (novel).

Figure 3. We randomly select 200 base classes and 200 novel classes from LVIS dataset and use t-SNE to visualize the class embedding generated by our DetPro and the classical prompt engineering. (a) base class embedding generated by prompt engineering; (b) base class embedding generated by DetPro; (c) novel class embedding generated by prompt engineering; (d) novel class embedding generated by DetPro. Each point denotes a category. Class embedding generated by our method is more discriminative in the embedding space, which attributes to the involvement of background proposals during training.

6. Conclusion

In this paper, we propose a novel method named detection prompt (DetPro), aiming to learn continuous prompt representations for open-vocabulary object detection based on the pre-trained vision-language model. Different from the previous classification-oriented prompt learning method, DetPro presents a background interpretation scheme to include negative proposals in images into the training, and a context grading scheme to separate positive proposals in image foreground for tailored prompt training. We assemble DetPro with ViLD and conduct a series of studies to demonstrate the importance of involving both proposals in both foreground and background in prompt representation learning for open-vocabulary object detection. Experiments on LVIS and transfer learning on Pascal VOC, COCO, Objects365 demonstrate the effectiveness and generalization ability of our approach.

Acknowledgements

This work was partially supported by National Nature Science Foundation of China (No. 61836004, 61828602), National Key Research and Development Program of China (Grant No. 2021ZD0200300), National Key R&D Program of China (2018AAA0102600), and Beijing Academy of Artificial Intelligence (BAAI).

Method	AP_r	AP_c	AP_f	AP
ViLD-text* [7]	12.1	24.2	28.9	23.9
DetPro-text	**14.2**	23.9	28.9	24.2

Table 9. We compare our DetPro-text with the ViLD-text. * denotes our re-implementation version, see Section 5.2 for the details.

Positive samples (%)	AP_r	AP_c	AP_f	AP
10	18.2	25.4	28.2	25.3
30	18.4	25.1	28.2	25.1
50	18.8	25.4	28.2	25.4
100	**19.1**	**25.4**	**28.2**	**25.4**

Table 10. Ablation study of using different number of positive samples for DetPro training.

A. More Experiments and Analysis

A Variant of DetPro. Following ViLD [7], we present a variant of DetPro named DetPro-text, and compare it with ViLD-text. In the DetPro-text, we remove the image head and only use a text head for training and inference. We use the LVIS setting as described in Section 5.3. Table 9 shows the comparison.

Using Different Number of Positive Samples for Training. We also study the effects of using different number of positive samples in DetPro training as shown in Table 10. The LVIS setting is adopted in this study. We observe that using all positive samples results in the best generalization performance on novel classes.

Accuracy of Proposal Classification. In our DetPro, we first optimize the prompt representations then feed them into CLIP text encoder to generate class embedding as classifiers of the detector. Here we report the image proposal classification accuracy on the LVIS dataset to demonstrate the effectiveness of our approach. Concretely, given a set of proposals generated by RPN, we resize each proposal to the size of 224×224 and feed it into the CLIP image encoder to extract its image embedding, then we compute the similarities between the image embedding and all class embedding to predict its class. We compare our approach with the prompt engineering. Table 11 and Table 12 show top-1 and top-5 accuracy, respectively. Remarkable improvements are observed on both base classes and novel classes, indicating that the prompt representations learned by our DetPro are also beneficial to the open-vocabulary image classification task.

Assembling the Well-trained ViLD with DetPro. In Section 4.3 of the main paper, we use class embedding generated by DetPro for the ViLD training and inference. In this study, we first use class embeddings generated by prompt engineering as classifiers of the detector to train ViLD, and assemble the well-trained ViLD with our DetPro for inference, by simply replacing the original class embedding in the image head with the ones generated by our DetPro. It can be seen in Table 13 that simply assembling the original ViLD with DetPro trained with different ensemble strategies (see Table 6 of the main paper) already shows non-negligible improvements on novel classes.

Method	Base class	Novel class
Prompt engineering	20.1	17.7
DetPro	**24.4**	**21.7**

Table 11. Top-1 accuracy of proposal classification.

Method	Base class	Novel class
Prompt engineering	39.3	37.2
DetPro	**49.0**	**40.3**

Table 12. Top-5 accuracy of proposal classification.

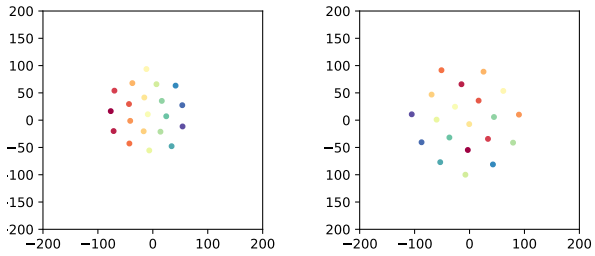
Method	AP_r	AP_c	AP_f	AP
ViLD*	16.8	25.6	28.5	25.2
DetPro-Ensemble(0.5:1.0:0.1)	18.1	25.7	28.3	25.4
DetPro-Ensemble(0.6:1.0:0.1)	18.0	25.4	28.2	25.2
DetPro-Ensemble(0.7:1.0:0.1)	18.0	25.4	28.2	25.3
DetPro-Ensemble(0.8:1.0:0.1)	17.9	25.7	28.3	25.4

Table 13. Assembling the well-trained ViLD with DetPro trained under different settings outperforms the original ViLD.

T-SNE Visualization for Transferred Datasets. In Section 5.5, we generate the class embedding for the LVIS dataset and show the t-SNE figure. Here we use t-SNE to visualize the class embedding generated by our DerPro and prompt engineering on transferred datasets including Pascal VOC, COCO, and Objects365. Figure 4-6 show the comparison. We observe the same phenomenon that the class embedding generated by DetPro is more discriminative in the embedding space, which further validates their suitability serving as the region classifiers for open-vocabulary object detection.

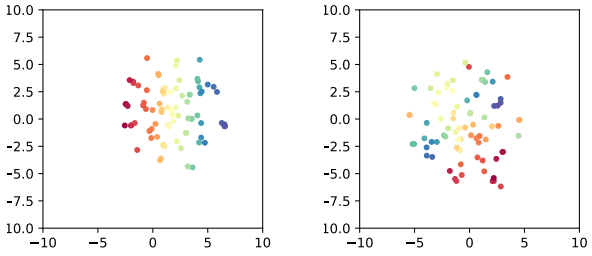
B. More Implementation Details

More Details of Our Open-world Object Detector. We use multi-scale training with the size of (1333, 640), (1333, 672), (1333, 704), (1333, 736), (1333, 768), (1333, 800). For RPN, we apply an NMS with a threshold of 0.7 and generate a maximum of 1000 proposals. We apply a class-agnostic NMS with a threshold of 0.5 on the final predictions and set the maximum number of output bounding



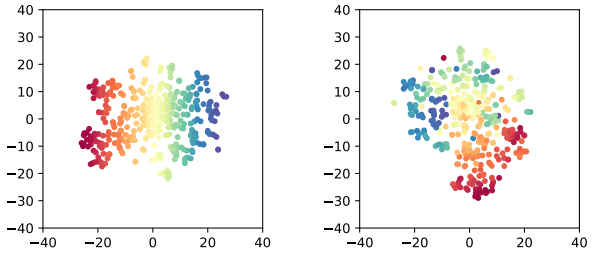
(a) Prompt engineering. (b) DetPro.

Figure 4. T-SNE visualization for Pascal VOC dataset.



(a) Prompt engineering. (b) DetPro.

Figure 5. T-SNE visualization for COCO dataset.



(a) Prompt engineering. (b) DetPro.

Figure 6. T-SNE visualization for Objects365 dataset.

boxes to 300.

More Details of DetPro Training. We set the batch size as 512. We use a cross-entropy loss with a temperature parameter of 0.01.

References

[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 2

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Global context networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[3] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *Advances in Neural Information Processing Systems*, 33:13564–13574, 2020. 2

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[5] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 2

[6] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[7] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1, 2, 5, 6, 7, 9

[8] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 1, 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[10] 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. *arXiv preprint arXiv:2102.05918*, 2021. 1, 2

[11] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[13] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020. 1

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[15] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[16] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2

[17] Xiang Ming, Fangyun Wei, Ting Zhang, Dong Chen, and Fang Wen. Group sampling for scale invariant face detection.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3456, 2019. 2

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1, 2, 3, 5

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2, 5

[21] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[22] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014. 2

[23] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8430–8439, 2019. 6

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[25] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017. 2

[26] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[27] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[28] Jernej Ule, Kirk B Jensen, Matteo Ruggiu, Aldo Mele, Aljaž Ule, and Robert B Darnell. Clip identifies nova-regulated rna networks in the brain. *Science*, 302(5648):1212–1215, 2003. 2

[29] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018. 2

[30] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37, 2019. 2

[31] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 6

[32] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*, pages 527–544. Springer, 2020. 2

[33] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018. 2

[34] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021. 2

[35] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Stephen Lin, Han Hu, and Xiang Bai. Bootstrap your object detector via mixed training. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[36] Yukuan Yang, Fangyun Wei, Miaojing Shi, and Guoqi Li. Restoring negative information in few-shot object detection. In *Advances in Neural Information Processing Systems*, 2020. 2

[37] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021. 2

[38] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 2, 3, 7, 8

[39] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4):998–1010, 2019. 2