

# Detecting Twenty-thousand Classes using Image-level Supervision

Xingyi Zhou<sup>1,2\*</sup> Rohit Girdhar<sup>1</sup> Armand Joulin<sup>1</sup> Phillip Krähenbühl<sup>2</sup> Ishan Misra<sup>1</sup>

<sup>1</sup>Meta AI

<sup>2</sup>The University of Texas at Austin

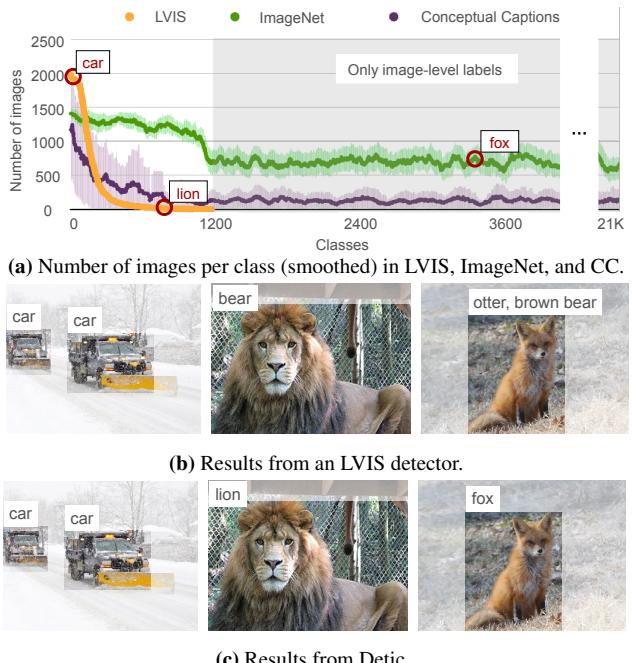
## Abstract

Current object detectors are limited in vocabulary size due to the small scale of detection datasets. Image classifiers, on the other hand, reason about much larger vocabularies, as their datasets are larger and easier to collect. We propose Detic, which simply trains the classifiers of a detector on image classification data and thus expands the vocabulary of detectors to tens of thousands of concepts. Unlike prior work, Detic does not assign image labels to boxes based on model predictions, making it much easier to implement and compatible with a range of detection architectures and backbones. Our results show that Detic yields excellent detectors even for classes without box annotations. It outperforms prior work on both open-vocabulary and long-tail detection benchmarks. Detic provides a gain of 2.4 mAP for all classes and 8.3 mAP for novel classes on the open-vocabulary LVIS benchmark. On the standard LVIS benchmark, Detic reaches 41.7 mAP for all classes and 41.7 mAP for rare classes. For the first time, we train a detector with all the twenty-one-thousand classes of the ImageNet dataset and show that it generalizes to new datasets without finetuning. Code is available at <https://github.com/facebookresearch/Detic>.

## 1. Introduction

Object detection consists of two sub-problems - finding the object (localization) and naming it (classification). Traditional methods tightly couple these two sub-problems and thus rely on box labels for all classes. Despite many data collection efforts, detection datasets [16, 25, 30, 44] are much smaller in overall size and object classes (vocabulary) than image classification datasets [9]. For example, the recent LVIS detection dataset [16] has 1000+ classes with 120K images; OpenImages [25] has 1.8M images for 500 classes. Moreover, not all classes contain sufficient annotations to train a robust detector (see Figure 1b). In classification, even the ten-year-old ImageNet dataset [9] has 21K classes and 14M images (Figure 1a).

In this paper, we propose **Detector with image classes**

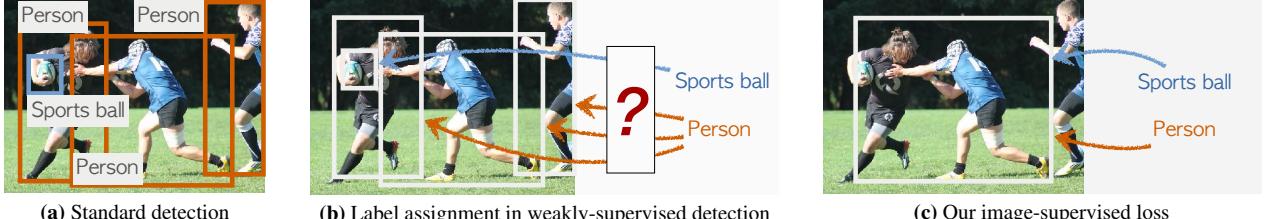


**Figure 1.** Qualitative results from a strong open-vocabulary LVIS detector (middle) and our open-vocabulary detector trained with image-level supervision (bottom). Our **Detector with image classes** (Detic) leverages image-level supervision (top) with a simple loss and improves detection performance for all classes.

(Detic) that uses image-level supervision in addition to detection supervision. We observe that the localization and classification sub-problems can be decoupled. Modern region proposal networks already localize many ‘new’ objects using existing detection supervision. Thus, we focus on the classification sub-problem and use image-level labels to train the classifier and broaden the vocabulary of the detector. We propose a simple classification loss that applies the image-level supervision to the proposal with the largest spatial size, and do not supervise other outputs for image-labeled data. This is easy to implement and massively expands the detector’s vocabulary.

Most existing weakly-supervised detection techniques [12, 20, 32, 53, 61] use the weakly labeled data to supervise *both* the localization and classification sub-problems of detection. Since image-classification data has no box labels, these methods develop various label-to-box

\*Work done during an internship at Meta.



**Figure 2.** **(left)** Standard detection requires ground-truth labeled boxes and cannot leverage image-level labels. **(center)** Weakly supervised methods [3, 39, 40] use image-level labels by assigning them to the detector’s predicted boxes (proposals). Unfortunately, this assignment is error-prone especially for large vocabulary detection. **(right)** Detic omits label-to-box assignment and supervise all image-labels to the *max-size* proposal. We show that this loss is both simpler and performs better than prior work.

assignment techniques to obtain boxes. For example, YOLO9000[40] and DLWL[39] assign the image-label to proposals with high prediction scores. Unfortunately, this assignment requires good initial detections which leads to a chicken-and-egg problem—we need a good detector for good label assignment, but we need many boxes to train a good detector. Our method completely side-steps the label assignment process by supervising the classification sub-problem alone when using classification data. This also enables our method to learn detectors for new classes which would have been impossible to predict and assign.

Experiments on the open-vocabulary LVIS [15, 16] and the open-vocabulary COCO [2] benchmarks show that our method can significantly improve over a strong box-supervised baseline, on both novel and base classes. With image-level supervision from ImageNet-21K [9], our model trained without novel class detection annotations improves the baseline by 8.3 point and matches the performance of using full class annotations in training. With the standard LVIS annotations, our model reaches 41.7 mAP and 41.7 mAP<sub>rare</sub>, narrowing the gap between rare classes and all classes to 0. On open-vocabulary COCO, our method outperforms the previous state-of-the-art OVR-CNN [66] by 5 point with the same detector and data. Finally, we train a detector using the full ImageNet-21K with more than twenty-thousand classes. Our detector generalizes much better to new datasets [25, 44] with disjoint label spaces, reaching 21.5 mAP on Objects365 and 55.2 mAP50 on OpenImages, without seeing any images from the corresponding training sets.

## 2. Related Work

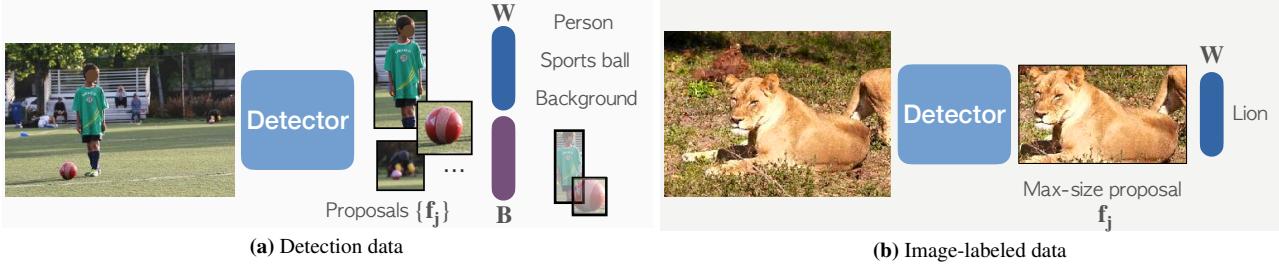
**Weakly-supervised object detection (WSOD)** trains object detector using image-level labels. Many works use only image-level labels without any box supervision [26, 46, 47, 57, 64]. WSDDN [3] and OIRC [54] use a subnetwork to predict per-proposal weighting and sum up proposal scores into a single image scores. PCL [53] first clusters proposals and then assign image labels at the cluster level. CASD [20] further introduces feature-level attention and self-distillation. As no bounding box supervision is used in training, these methods rely on low-level region proposal techniques [1, 56],

which leads to reduced localization quality.

Another line of WSOD work uses bounding box supervision together with image labels, known as **semi-supervised WSOD** [11, 12, 28, 31, 55, 62, 69]. YOLO9000 [40] mixes detection data and classification data in the same mini-batch, and assigns classification labels to anchors with the highest predicted scores. DLWL [39] combines self-training and clustering-based WSOD [53], and again assigns image labels to max-scored proposals. MosaicOS [67] handles domain differences between detection and image datasets by mosaic augmentation [4] and proposed a three-stage self-training and finetuning framework. In segmentation, Pinheiro *et al.* [36] use a log-sum-exponential function to aggregate pixels scores into a global classification. Our work belongs to semi-supervised WSOD. Unlike prior work, we use a simple image-supervised loss. Besides image labels, researchers have also studied complementary methods for weak localization supervision like points [7] or scribbles [42].

**Open-vocabulary object detection**, or also named **zero-shot object detection**, aims to detect objects outside of the training vocabulary. The basic solution [2] is to replace the last classification layer with language embeddings (e.g., GloVe [35]) of the class names. Rahman *et al.* [38] and Li *et al.* [29] improve the classifier embedding by introducing external text information. OVR-CNN [66] pretrains the detector on image-text pairs using contrastive learning. ViLD [15] upgrades the language embedding to CLIP [37] and distills region features from CLIP image features. Our work also uses CLIP [37] embeddings as the classifier, but does not use distillation. Instead, we incorporate additional image-labeled data for co-training.

**Large-vocabulary object detection** [16] requires detecting 1000+ classes. Many existing works focus on handling the long-tail problem [6, 13, 27, 34, 59, 68]. Repeat factor sampling (RFS) [16] oversamples classes with fewer annotations. Equalization losses [49, 50] and SeeSaw loss [58] reweights the per-class loss by balancing the gradients [49] or number of samples [58]. Federated Loss [70] substitutes classes per-iteration to mimic the federated annotation [16]. Yang *et al.* [63] detects 11K classes with a label hierarchy. Our method builds on these advances, and we tackle the problem from a different aspect: using additional image-labeled data.



**Figure 3. Approach Overview.** We mix train on detection data and image-labeled data. When using detection data, our model uses the standard detection losses to train the classifier ( $\mathbf{W}$ ) and the box prediction branch ( $\mathbf{B}$ ) of a detector. When using image-labeled data, we only train the classifier using our modified classification loss. Our classification loss trains the features extracted from the largest-sized proposal predicted by the network. We use CLIP embeddings [37] as the classification weights ( $\mathbf{W}$ ) for open-vocabulary detection.

**Language supervision for object detection** is a recent popular research topic. VirTex [10] and OVR-CNN [66] pretrain backbones on language tasks and show benefits for detection. Cap2Det [65] learns a mapping from sentences to image labels in the detector’s vocabulary then applies WSOD [3]. MDETR [23] performs instance-level detection by training on paired object-caption annotations. We use language data [45] in a similar way as Cap2Det, but simpler: we extract image labels from captions using a naive text-match.

### 3. Approach

We train object detectors using both object detection and image classification datasets. We propose a simple way to leverage image supervision to learn object detectors, including for classes without box labels. We first describe the object detection problem and then detail our approach.

#### 3.1. Notation and Preliminaries

Given an image  $\mathbf{I} \in \mathbb{R}^{3 \times h \times w}$ , object detection solves the two subproblems of (1) localization: find all objects with their location, represented as a box  $\mathbf{b}_j \in \mathbb{R}^4$  and (2) classification: assign a class label  $c_j \in \mathcal{C}^{\text{test}}$  to the  $j$ -th object. Here  $\mathcal{C}^{\text{test}}$  is the class vocabulary provided by the user at test time. During training, we use a detection dataset  $\mathcal{D}^{\text{det}} = \{(\mathbf{I}, \{(\mathbf{b}, c)_k\})_i\}_{i=1}^{|\mathcal{D}^{\text{det}}|}$  with vocabulary  $\mathcal{C}^{\text{det}}$  that has both class and box labels. We also use an image classification dataset  $\mathcal{D}^{\text{cls}} = \{(\mathbf{I}, \{c_k\})_i\}_{i=1}^{|\mathcal{D}^{\text{cls}}|}$  with vocabulary  $\mathcal{C}^{\text{cls}}$  that only has image-level class labels. The vocabularies  $\mathcal{C}^{\text{test}}, \mathcal{C}^{\text{det}}, \mathcal{C}^{\text{cls}}$  may or may not overlap.

**Traditional Object detection** considers  $\mathcal{C}^{\text{test}} = \mathcal{C}^{\text{det}}$  and  $\mathcal{D}^{\text{cls}} = \emptyset$ . Predominant object detectors [18, 41] follow a two-stage framework. The first stage, called the *region proposal network* (RPN), takes the image  $\mathbf{I}$  and produces a set of object proposals  $\{(\mathbf{b}, \mathbf{f}, o)_j\}$ , where  $\mathbf{f}_j \in \mathbb{R}^D$  is a  $D$ -dimensional region feature and  $o \in \mathbb{R}$  is the objectness score. The second stage takes the object feature and outputs a classification score and a refined box location for each object,  $s_j = \mathbf{W}\mathbf{f}_j$ ,  $\hat{\mathbf{b}}_j = \mathbf{B}\mathbf{f}_j + \mathbf{b}_j$ , where  $\mathbf{W} \in \mathbb{R}^{|\mathcal{C}^{\text{det}}| \times D}$  and  $\mathbf{B} \in \mathbb{R}^{4 \times D}$  are the learned weights of the classification layer

and the regression layer, respectively<sup>1</sup>. Our work focuses on improving classification in the second stage. We observe that the proposal network of a detector and the bounding box regressors can already generalize to new classes (see Appendix A). This allows us to focus on training the classifier. **Open-vocabulary object detection** allows  $\mathcal{C}^{\text{test}} \neq \mathcal{C}^{\text{det}}$ . Simply replacing the classification weights  $\mathbf{W}$  with language embeddings of class names converts a traditional detector to an open-vocabulary detector [2]. We follow Gu *et al.* [15] to use the CLIP embeddings [37] as the classification weights. CLIP [37] trains aligned image and text features on a large corpus of image-text pairs and has demonstrated great zero-shot generalization ability in image classification. In theory, this open-vocabulary detector can detect any object class. However, in practice, it yields unsatisfying detection results as shown in Figure 1. Our method uses image-level supervision to improve object detection including in the open-vocabulary setting.

#### 3.2. Detic: Detector with Image Classes

As shown in Figure 3, our method leverages the box labels from detection datasets  $\mathcal{D}^{\text{det}}$  and image-level labels from classification datasets  $\mathcal{D}^{\text{cls}}$ . During training, we compose a mini-batch using images from both types of datasets. For images with box labels, we follow the standard two-stage detector training [41]. For image-level labeled images, we only train the features from a fixed region proposal for classification. Thus, we only compute the localization losses (RPN loss and bounding box regression loss) on images with ground truth box labels. Below we describe our modified classification loss for image-level labels.

**Classifier-only training with image labels.** A sample from the weakly labeled dataset  $\mathcal{D}^{\text{cls}}$  contains an image  $\mathbf{I}$  and a set of  $K$  labels  $\{c_k\}_{k=1}^K$ . We use the region proposal network to extract  $N$  object features  $\{(\mathbf{b}, \mathbf{f}, o)_j\}_{j=1}^N$ . We propose simple ways to use the image labels  $\{c_k\}_{k=1}^K$  and supervise the region proposal features.

Our first idea is to use the whole image as a new “proposal”

<sup>1</sup>We omit the two additional linear layers and the bias in the second stage for notation simplicity.

box. We call this loss **image-box**. We ignore all proposals from the RPN, and instead use an injected box of the whole image  $\mathbf{b}' = (0, 0, w, h)$ . We then apply the classification loss to its RoI features  $\mathbf{f}'$  for all classes  $c \in \{c_k\}_{k=1}^K$ :

$$L_{\text{image-box}} = BCE(\mathbf{W}\mathbf{f}', c)$$

where  $BCE(s, c) = -\log\sigma(s_c) - \sum_{k \neq c} \log(1 - \sigma(s_k))$  is the binary cross-entropy loss, and  $\sigma$  is the sigmoid activation. Thus, our loss uses the features from the same ‘proposal’ for solving the classification problem for all the classes  $\{c_k\}$ .

In practice, the image-box can be replaced by smaller boxes. We introduce two alternatives: the proposal with the **max object score** or the proposal with the **max size**:

$$L_{\text{max-object-score}} = BCE(\mathbf{W}\mathbf{f}_j, c), j = \operatorname{argmax}_j o_j$$

$$L_{\text{max-size}} = BCE(\mathbf{W}\mathbf{f}_j, c), j = \operatorname{argmax}_j (\text{size}(\mathbf{b}_j))$$

We show that all these three losses can effectively leverage the image-level supervision, while the max-size loss performs the best. We thus use the max-size loss by default for image-supervised data. We also note that the classification parameters  $\mathbf{W}$  are shared across both detection and classification data, which greatly improves detection performance (§ 4.6). The overall training objective is

$$L(\mathbf{I}) = \begin{cases} L_{\text{rpn}} + L_{\text{reg}} + L_{\text{cls}}, & \text{if } \mathbf{I} \in \mathcal{D}^{\text{det}} \\ \lambda L_{\text{max-size}}, & \text{if } \mathbf{I} \in \mathcal{D}^{\text{cls}} \end{cases}$$

where  $L_{\text{rpn}}$ ,  $L_{\text{reg}}$ ,  $L_{\text{cls}}$  are standard losses in a two-stage detector,  $\lambda = 0.1$  is the weight of our loss.

**Relation to weakly-supervised detection.** In traditional weakly-supervised detection [3, 39, 40], a popular idea is to assign the image to the proposals. Let  $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_N)$  be the stacked feature of all object proposals and  $\mathbf{S} = \mathbf{WF}$  be their classification scores. For each  $c \in \{c_k\}_{k=1}^K$ ,  $L = BCE(\mathbf{S}_j, c), j = \mathcal{F}(\mathbf{S}, c)$ , where  $\mathcal{F}$  is the label-to-box assignment process. In most methods,  $\mathcal{F}$  is a function of the prediction  $\mathbf{S}$ . For example,  $\mathcal{F}$  selects the proposal with max score on  $c$ . Our key insight is that  $\mathcal{F}$  should *not* depend on the prediction  $\mathbf{S}$ . In large-vocabulary detection, the initial recognition ability of rare or novel classes is low, making the label assignment process inaccurate. Our method side-steps this prediction-and-assignment process entirely and relies on a fixed supervision criteria.

## 4. Experiments

We evaluate Detic on the large-vocabulary object detection dataset LVIS [16]. We mainly use the open-vocabulary setting proposed by Gu *et al.* [15], and also report results on the standard LVIS setting. Additionally, we compare to prior work on the popular open-vocabulary COCO benchmark [2]. We describe our LVIS setups below and provide full details of the COCO benchmark in Appendix B.

**LVIS.** The LVIS [16] dataset has object detection and instance segmentation labels for 1203 classes with 100K images. The classes are divided into three groups - frequent, common, rare based on the number of training images. We refer to this standard LVIS training set as *LVIS-all*. Following ViLD [15], we remove the labels of 317 rare-class from training and consider them as novel classes in testing. We refer to this partial training set with only frequent and common classes as *LVIS-base*. We report mask mAP which is the official metric for LVIS. While our model is developed for box detection, we use a standard class-agnostic mask head [18] to produce segmentation masks for boxes. We train the mask head only on detection data.

**Image-supervised data.** We use two sources of image-supervised data: ImageNet-21K [9] and Conceptual Captions [45]. ImageNet-21K (IN-21K) contains 14M images for 21K classes. For ease of training and evaluation, most of our experiments use the 997 classes that overlap with the LVIS vocabulary and denote this subset as IN-L. Conceptual Captions [45] (CC) is an image captioning dataset containing 3M images. We extract image labels from the captions using exact text-matching and keep images whose captions mention at least one LVIS class. See Appendix C for results of directly using captions. The resulting dataset contains 1.5M images with 992 LVIS classes. We summarize the datasets used in our experiments below.

Notation	Definition	#Imgs	#Cls
LVIS-all	The original LVIS dataset [16]	100K	1203
LVIS-base	LVIS without rare-class annotations	100K	886
IN-21K	The original ImageNet-21K dataset [9]	14M	21k
IN-L	997 overlapping IN-21K classes with LVIS	1.2M	997
CC	Conceptual Captions [45] with LVIS classes	1.5M	992

### 4.1. Implementation details

**Box-Supervised: a strong LVIS baseline.** We first establish a strong baseline on LVIS to demonstrate that our improvements are orthogonal to recent advances in object detection. The baseline only uses the supervised bounding box labels. We use the CenterNet2 [70] detector with ResNet50 [19] backbone. We use Federated Loss [70] and repeat factor sampling [16]. We use large scale jittering [14] with input resolution  $640 \times 640$  and train for a  $4 \times$  ( $\sim 48$  LVIS epochs) schedule. To show our method is compatible with better pretraining, we use ImageNet-21k pretrained backbone weights [43]. As described in § 3.1, we use the CLIP [37] embedding as the classifier. Our baseline is 9.1 mAP higher than the detectron2 baseline [60] (31.5 vs. 22.4 mAP<sup>mask</sup>) and trains in a similar time (17 vs. 12 hours on 8 V100 GPUs). See Appendix D for more details.

**Resolution change for image-labeled images.** ImageNet images are inherently smaller and more object-focused than LVIS images [67]. In practice, we observe it is important

#		LVIS-base + IN-L		LVIS-base + CC		Open-vocabulary COCO	
		mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP50 <sup>box</sup> <sub>all</sub>	mAP50 <sup>box</sup> <sub>novel</sub>
1	Box-Supervised (base class)	$30.0 \pm 0.4$	$16.3 \pm 0.7$	$30.0 \pm 0.4$	$16.3 \pm 0.7$	39.3	1.3
2	Box-Supervised (base class, finetuned)	$29.7 \pm 0.5$	$15.7 \pm 1.0$	$29.7 \pm 0.5$	$15.7 \pm 1.0$	40.6	1.0
3	Self-training [48]	$30.3 \pm 0.0$	$15.6 \pm 0.1$	$30.1 \pm 0.2$	$15.9 \pm 0.8$	39.5	1.8
4	Self-training [48] with image labels	$31.7 \pm 0.3$	$19.4 \pm 0.7$	$30.7 \pm 0.1$	$18.2 \pm 0.6$	39.2	0.9
5	WSDDN [3]	$29.8 \pm 0.2$	$15.6 \pm 0.3$	$30.0 \pm 0.1$	$16.5 \pm 0.8$	39.9	5.9
6	DLWL* [39]	$30.6 \pm 0.1$	$18.2 \pm 0.2$	$29.7 \pm 0.3$	$16.9 \pm 0.6$	42.9	19.6
7	Predicted [40]	$31.2 \pm 0.3$	$20.4 \pm 0.9$	$29.4 \pm 0.1$	$15.9 \pm 0.6$	41.9	18.7
8	Detic (Ours)	<b><math>32.4 \pm 0.1</math></b>	<b><math>24.6 \pm 0.3</math></b>	<b><math>30.9 \pm 0.2</math></b>	<b><math>19.5 \pm 0.3</math></b>	<b>44.7</b>	<b>24.1</b>
9	Box-Supervised (all class)	$31.1 \pm 0.4$	$25.5 \pm 0.7$	$31.1 \pm 0.4$	$25.5 \pm 0.7$	54.9	60.0

**Table 1. Different ways to use image supervision.** We show overall and novel-class mAP on open-vocabulary LVIS [15] (middle two columns, with 886 base classes and 317 novel classes) and open-vocabulary COCO [2](right column, with 48 base classes and 17 novel classes). The LVIS models are trained using our strong baseline § 4.1. The COCO models are trained using ResNet50-C4 [2]. All models are finetuned on the model in row 1. Rows 1-2: supervised training on base-classes; It has non-zero novel-class mAP as it uses the CLIP classifier. Rows 3-4: self-training; Rows 5-7: prediction-based WSOD; Row 8: Our simple image-supervised WSOD. Row 9: The top-line detector trained with novel-class annotations. We repeat LVIS experiments for 3 runs and report mean/ std.

to use smaller image resolution for ImageNet images. Using smaller resolution in addition allows us to increase the batch-size with the same computation. In our implementation, we use  $320 \times 320$  for ImageNet and CC and ablate this in Appendix E.

**Multi-dataset training.** We sample detection and classification mini-batches in a  $1 : 1$  ratio, regardless of the original dataset size. We group images from the same dataset on the same GPU to improve training efficiency [71].

**Training schedules.** To shorten the experimental cycle and have a good initialization for prediction-based WSOD losses [39, 40], we always first train a converged base-class-only model ( $4 \times$  schedule) and finetune on it with additional image-labeled data for another  $4 \times$  schedule. The  $4 \times$  schedule for our joint training consists of  $\sim 24$  LVIS epochs plus  $\sim 4.8$  ImageNet epochs or  $\sim 3.8$  CC epochs. Training our ResNet50 model takes  $\sim 22$  hours on 8 V100 GPUs. The large 21K Swin-B model trains in  $\sim 24$  hours on 32 GPUs.

## 4.2. Open-vocabulary detection with image labels

We first explore how to use image-level supervision for open-vocabulary detection. We experiment under three settings: LVIS-base with IN-L [9], LVIS-base with CC [45], and open-vocabulary COCO with COCO-caption as the image-labeled data. These three settings differ in numbers of objects and numbers of image-labels per images<sup>2</sup>. Table 1 shows the results. The baselines (Box-Supervised (base class)) are trained without access to novel class bounding box labels. It uses the CLIP classifier [15] and has open-vocabulary capabilities and obtains  $15.4$  mAP<sub>novel</sub>. We first show our improvements are not trivially from the longer

<sup>2</sup>COCO images are designed to contain many objects; ImageNet images mostly contain one object; CC does not have #objects preferences by design.

training iterations in finetuning. Table 1 row 2 shows further finetuning the model using box-supervision does not improve the performance. We compare with the following ways to use image-level supervision:

**Self-training** applies the Box-Supervised model to the image-classification data and collects all predictions with score  $> 0.5$  as pseudo-labels for finetuning. This self-training method [17, 48] is originally proposed for unlabeled images and does not use image labels. We improve it by filtering out pseudo-labels that do not match the image labels. The results in Table 1 rows 3, 4 show self-training with image labels is effective on LVIS. However, self-training does not improve on COCO as the base model is too weak to generate meaningful novel-class pseudo labels.

**Weakly-supervised Object Detection (WSOD)** relies on assigning image-labels to the predicted boxes. In Table 1 rows 5-7, we compare to these methods (See Appendix F for implementation details). For DLWL [39], we implement a simplified version that does not include bootstrapping and refer to it as DLWL\*. On the LVIS dataset, WSOD losses show an improvement while using ImageNet but do not provide gains with CC. We believe that CC images contain more objects which makes the assignment problem harder.

**Detic (Ours).** Table 1 row 8 shows the results of our Detector with Image Classes. Our simple loss significantly improves the baseline and other alternatives, under all the three settings. On the novel classes, Detic provides a gain of 8.3 points with ImageNet and 3.2 points with CC. Thus, Detic with image-level labels leads to strong open-vocabulary detection performance and can provide orthogonal gains to existing open-vocabulary detectors [2].

To further understand the open-vocabulary capabilities of Detic, we also report the *top-line* results trained with box

	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP <sup>mask</sup> <sub>c</sub>	mAP <sup>mask</sup> <sub>f</sub>
ViLD-text [15]	24.9	10.1	23.9	<b>32.5</b>
ViLD [15]	22.5	16.1	20.0	28.3
ViLD-ens. [15]	25.5	16.6	24.6	30.3
Detic	<b>26.8</b>	<b>17.8</b>	<b>26.3</b>	31.6

**Table 2. Open-vocabulary LVIS compared to ViLD [15].** We train our model *using their training settings and architecture* (MaskRCNN-ResNet50, training from scratch). We report mask mAP and its breakdown to novel (rare), common, frequent classes. Variants of ViLD use distillation (ViLD) or ensembling (ViLD-ens.). Detic uses a single model and improves both mAP and mAP<sub>novel</sub>.

labels for all classes denoted as Box-Supervised (all class) (row 9). Despite not using box labels for the novel classes, Detic with ImageNet performs favorably compared to Box-Supervised (all class). This result also suggests that bounding box annotations may not be required for new classes. Our Detic method combined with large image classification datasets is an easy and effective alternative for increasing detector vocabulary. We provide more comparisons to WSOD methods in Appendix G.

### 4.3. Comparison to open-vocabulary detectors

We now use Detic to train open-vocabulary object detectors and compare them to state-of-the-art methods.

**Open-vocabulary LVIS.** We compare to ViLD [15], which first uses CLIP embeddings [37] for open-vocabulary detection. We strictly follow their training setup and model architecture (Appendix H) and report results in Table 2. Here ViLD-text is exactly our Box-Supervised baseline. Detic provides a gain of 7.7 points on mAP<sub>novel</sub>. Compared to ViLD-text, ViLD, which uses knowledge distillation from the CLIP visual backbone, improves mAP<sub>novel</sub> at the cost of hurting overall mAP. Ensembling the two models, ViLD-ens provides improvements for both metrics. On the other hand,

	mAP50 <sup>box</sup> <sub>all</sub>	mAP50 <sup>box</sup> <sub>novel</sub>	mAP50 <sup>box</sup> <sub>base</sub>
Base-only†	39.9	0	<b>49.9</b>
Base-only (CLIP)	39.3	1.3	48.7
WSDDN [3]†	24.6	20.5	23.4
Cap2Det [65]†	20.1	20.3	20.1
SB [2]‡	24.9	0.31	29.2
DELO [72]‡	13.0	3.41	13.8
PL [38]‡	27.9	4.12	35.9
OVR-CNN [66]†	39.9	22.8	46.0
Detic	<b>45.0</b>	<b>27.8</b>	47.1

**Table 3. Open-vocabulary COCO [2].** We compare Detic using the same training data and architecture from OVR-CNN [66]. We report bounding box mAP at IoU threshold 0.5 using Faster R-CNN with ResNet50-C4 backbone. Detic builds upon the CLIP baseline (second row) and shows significant improvements over prior work. †: results quoted from OVR-CNN [66] paper or code. ‡: results quoted from the original publications.

	Objects365 [44]		OpenImages [25]	
	mAP <sup>box</sup>	mAP <sup>box</sup> <sub>rare</sub>	mAP50 <sup>box</sup>	mAP50 <sup>box</sup> <sub>rare</sub>
Box-Supervised	19.1	14.0	46.2	61.7
Detic w. IN-L	21.2	17.8	53.0	67.1
Detic w. IN-21k	<b>21.5</b>	<b>20.0</b>	<b>55.2</b>	<b>68.8</b>
Dataset-specific oracles	31.2	22.5	69.9	81.8

**Table 4. Detecting 21K classes across datasets.** We use Detic to train a detector and evaluate it on multiple datasets *without re-training*. We report the bounding box mAP on Objects365 and OpenImages. Compared to the Box-Supervised baseline (trained on LVIS-all), Detic leverages image-level supervision to train robust detectors. The performance of Detic is 70%-80% of dataset-specific models (bottom row) that use dataset specific box labels.

Detic uses a single model which improves both novel and overall mAP, and outperforms the ViLD ensemble.

**Open-vocabulary COCO.** Next, we compare with prior works on open-vocabulary COCO benchmark [2] (see benchmark and implementation details in Appendix B & Appendix C). We strictly follow OVR-CNN [66] to use Faster R-CNN with ResNet50-C4 backbone and do not use any improvements from § 4.1. Following [66], we use COCO captions as the image-supervised data. We convert the captions into image labels and use both the image labels and captions as supervision.

Table 3 summarizes our results. As the training set contains only 48 base classes, the base-class only model (second row) yields low mAP on novel classes. Detic improves the baseline and outperforms OVR-CNN [66] by a large margin, using exactly the same model, training recipe, and data.

### 4.4. Detecting 21K classes across datasets

Next, we train a detector with the full 21K classes of ImageNet. We use our strong recipe with Swin-B [33] backbone. In practice, training a classification layer of 21K classes is computationally involved.<sup>3</sup> We adopt a modified Federated Loss [70] that uniformly samples 50 classes from the vocabulary at every iteration. We only compute classification scores and back-propagate on the sampled classes.

As there are no direct benchmark to evaluate detectors with such large vocabulary, we evaluate our detectors on new datasets *without finetuning*. We evaluate on two large-scale object detection datasets: Objects365v2 [44] and OpenImages [25], both with around 1.8M training images. We follow LVIS to split  $\frac{1}{3}$  of classes with the fewest training images as rare classes. Table 4 shows the results. On both datasets, Detic improves the Box-Supervised baseline by a large margin, especially on classes with fewer annotations. Using all the 21k classes further improves performance owing to the large vocabulary. Our single model significantly reduces the gap

<sup>3</sup>This is more pronounced in object detection than classification, as the “batch-size” for the classification layer is  $512 \times$  image-batch-size, where 512 is the number of RoIs per image.

	Backbone	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>r</sub>	mAP <sup>mask</sup> <sub>c</sub>	mAP <sup>mask</sup> <sub>f</sub>
Box-Supervised	ResNet50	31.5	25.6	30.4	35.2
Detic	ResNet50	<b>33.2</b>	<b>29.7</b>	<b>32.5</b>	<b>35.5</b>
Box-Supervised	Swin-B	40.7	35.9	40.5	<b>43.1</b>
Detic	Swin-B	<b>41.7</b>	<b>41.7</b>	<b>40.8</b>	42.6
MosaicOS [67]	ResNeXt-101	28.3	21.7	27.3	32.4
CenterNet2 [70]	ResNeXt-101	34.9	24.6	34.7	42.5
AsyncSLL [17]	ResNeSt-269	36.0	27.8	36.7	39.6
SeesawLoss [58]	ResNeSt-200	37.3	26.4	36.3	<b>43.1</b>
Copy-paste [14]	EfficientNet-B7	38.1	32.1	37.1	41.9
Tan et al. [51]	ResNeSt-269	38.8	28.5	39.5	42.7

**Table 5. Standard LVIS.** We evaluate our baseline (Box-Supervised) and Detic using different backbones on the LVIS dataset. We report the mask mAP. We also report prior work on LVIS using large backbone networks (single-scale testing). Detic improves over the baseline with increased gains for the rare classes. The improvements are consistent in the high-performance regime.

towards the dataset-specific oracles and reaches 70%-80% of their performance without using the corresponding 1.8M detection annotations. See Figure 4 for qualitative results.

#### 4.5. The standard LVIS benchmark

Finally, we evaluate Detic on the standard LVIS benchmark [16]. In this setting, the baseline (Box-Supervised) is trained with box and mask labels for all classes while Detic uses additional image-level labels from IN-L. In Table 5, we report the performance of these methods for two different backbones ResNet50 [19] and Swin-B [33]. We report the mask mAP across all classes and also split the metric for rare, common, and frequent classes. Detic brings consistent improvement over the baseline across these metrics for both the backbones. On the rare classes, Detic brings a significant improvement of  $\sim 4$  mAP<sub>r</sub> over the baseline for both backbones. With the Swin-B backbone, our model achieves 41.7 mAP and 41.7 mAP<sub>r</sub>, closing the gap between the overall mAP and the rare mAP. This suggests Detic effectively uses image-level labels to improve the performance of classes with very few boxes labels.

In Table 6, we compare to MosaicOS [67] which also uses image-level annotations to improve LVIS detectors. We strictly follow their training recipe (without any improvements in § 4.1) and report results using the vanilla Mask R-CNN [18] architecture. We train Detic using IN-L as the image-supervised data. MosaicOS uses IN-L and additional web-search images as image-supervised data. Detic outperforms MosaicOS [67] in mAP and mAP<sub>r</sub>, without using their multi-stage training framework and mosaic augmentation. See Appendix I for more detailed comparisons.

#### 4.6. Ablation studies

We now ablate our key components under the open-vocabulary LVIS setting with IN-L as the image-

	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>r</sub>	mAP <sup>mask</sup> <sub>c</sub>	mAP <sup>mask</sup> <sub>f</sub>
Box-Supervised	22.6	12.3	21.3	28.6
MosaicOS [67] <sup>†</sup>	24.5	18.3	<b>23.0</b>	<b>28.9</b>
Detic	<b>24.9</b>	<b>20.7</b>	<b>23.0</b>	28.7

**Table 6. Standard LVIS compared to MosaicOS.** We compare under the training settings of MosaicOS [67] and report the mask mAP. All methods use Mask R-CNN with the ResNet50. MosaicOS uses additional images searched from Google (indicated by <sup>†</sup>). Detic provides a greater improvement while being easier to train.

classification data. We use our strong training recipe as described in § 4.1 for all these experiments.

**Variants of Detic.** Table 7 (bottom) lists variants of our Detic loss (§ 3.2). All variants significantly improve over the Box-Supervised baseline. Amongst these variants, max-size proposal performs considerably better than others on open-vocabulary COCO. This is likely because COCO images have more objects, which makes the whole ‘image-box’ less representative of all the objects and ‘max-object-score’ arbitrary in picking one object. The mask-size proposal provides a focused region containing multiple objects.

**Shared classification parameters.** We study the importance of sharing the classification parameters  $\mathbf{W}$  across the image-supervised data and the detection data. In Table 7 (row 2), we consider Detic image-box but use a separate classification head (two linear layers) for image-supervised data. This variant can also be interpreted as multi-task learning with image-labeled data and improves mAP for both novel and all classes. However, it is less effective than Detic that shares the classifier parameters.

**Classifier weights.** We study the effect of different classifier weights  $\mathbf{W}$ . While our main open-vocabulary experiments use CLIP [37], we show the gain of Detic is independent of CLIP. We train Box-Supervised and Detic with different classifiers, including a standard random initialized and trained classifier, and other *fixed* language models [21, 22]. The results are shown in Table 8. By default, a trained classifier cannot recognize novel classes. However, Detic enables novel class recognition ability even in this setting

	LVIS-base+IN-L		LVIS-base+CC		ZS COCO	
	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP50 <sub>all</sub> <sup>box</sup>	mAP50 <sub>novel</sub> <sup>box</sup>
Box-Sup.	30.0	16.3	30.0	16.3	39.3	1.3
Multi-task	31.3	17.3	31.2	17.0	38.7	1.6
obj.-score	32.2	24.4	29.8	18.2	43.3	20.4
Image-box	<b>32.4</b>	23.8	30.9	19.5	43.4	21.0
Max-size	<b>32.4</b>	<b>24.9</b>	<b>31.0</b>	<b>19.8</b>	<b>44.7</b>	<b>24.1</b>

**Table 7. Variants of Detic.** We show overall mAP and novel class mAP on both open-vocabulary LVIS [15] and open-vocabulary COCO [2]. The experiment setting follows Table 1. Obj.-score is short for max-object-score. Max-size consistently performs the best among the variants and is our default choice.



**Figure 4. Qualitative results of our 21k-class detector.** We show random samples from images containing novel classes in OpenImages (top) and Objects365 (bottom) validation sets. We use the CLIP embedding of the corresponding vocabularies. We show LVIS classes in purple and novel classes in green. We use a score threshold of 0.5 and show the most confident class for each box. Best viewed on screen.

Classifier	Box-Supervised		Detic	
	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>
*CLIP [37]	30.2	16.4	32.4	24.9
Trained	27.4	0	31.7	17.4
FastText [22]	27.5	9.0	30.9	19.2
OpenCLIP [21]	27.1	8.9	30.7	19.4

**Table 8. Detic with different classifiers.** We vary the classifier used with Detic and observe that it works well with different design choices. While CLIP embeddings give the best overall performance (\* indicates our default), all classifiers benefit from our Detic.

	Pretrain data	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>
Box-Supervised	IN-1K	26.1	13.6
Detic	IN-1K	28.8 <small>(+2.7)</small>	21.7 <small>(+8.1)</small>
Box-Supervised	IN-21K	30.2	16.4
Detic	IN-21K	32.4 <small>(+2.2)</small>	24.9 <small>(+8.5)</small>

**Table 9. Detic with different pretraining data.** Top: our method using ImageNet-1K as pretraining and ImageNet-21K as co-training; Bottom: using ImageNet-21K for both pretraining and co-training. Co-training helps pretraining in both cases.

(17.4 mAP<sub>novel</sub> for classes without detection labels). Using language models such as FastText [22] or an open-source version of CLIP [21] leads to better novel class performance. CLIP [37] performs the best among them.

**Pretraining vs. Co-training.** Many existing methods use additional data only for pretraining [10, 66, 67], while we use image-labeled data for co-training. We present results of Detic with different types of pretraining in Table 9. Detic provides similar gains across different types of pretraining, suggesting that our gains are orthogonal to advances in pre-training. We believe that this is the case because pretraining improves the overall features, while Detic uses co-training which improves the classifier.

**Generalization to Deformable-DETR.** We apply Detic to the recent Transformer based Deformable-DETR [73] to

	mAP <sup>box</sup>	mAP <sup>box</sup> <sub>r</sub>	mAP <sup>box</sup> <sub>c</sub>	mAP <sup>box</sup> <sub>f</sub>
Box-Supervised	31.7	21.4	30.7	<b>37.5</b>
Detic	<b>32.5</b>	<b>26.2</b>	<b>31.3</b>	36.6

**Table 10. Detic applied to Deformable-DETR [73].** We report Box mAP on full LVIS. Our method improves Deformable-DETR.

study its generalization. We use their default training recipe, Federated Loss [70] and train for a 4× schedule ( $\sim 48$  LVIS epochs). We apply the image supervision to the query from the encoder with the max predicted size. Table 10 shows that Detic improves over the baseline (+0.8 mAP and +4.8 mAP<sub>r</sub>) and generalizes to Transformer based detectors.

## 5. Limitations and Conclusions

We present Detic which is a simple way to use image supervision in large-vocabulary object detection. While Detic is simpler than prior assignment-based weakly-supervised detection methods, it supervises all image labels to the same region and does not consider overall dataset statistics. We leave incorporating such information for future work. The generalization capabilities of Detic also benefit from the large-scale pretraining of CLIP, and it is an open question whether there are other ways to train the classifier. Moreover, open vocabulary generalization has no guarantees on extreme domains. Our experiments show Detic improves large-vocabulary detection with various weak data sources, classifiers, detector architectures, and training recipes. We hope Detic makes object detection easier to deploy and encourages future research in open-vocabulary detection.

**Ethical Considerations.** Our technical contribution is in leveraging image-labeled data for training large vocabulary detectors. We believe our technical contribution is neutral from an ethical standpoint, and our model needs further analysis before deployment. Detic’s wide range of detection capabilities may introduce similar challenges to many other visual recognition and open-set recognition methods [37]. As the user can define arbitrary detection classes, class design and semantics may impact the model output.

**Acknowledgments.** We thank Bowen Cheng and Ross Girshick for helpful discussions and feedback. This material is in part based upon work supported by the National Science Foundation under Grant No. IIS-1845485 and IIS-2006820. Xingyi is supported by a Facebook PhD Fellowship.

## References

- [1] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 2
- [2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 2, 3, 4, 5, 6, 7, 11
- [3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 2, 3, 4, 5, 6, 12
- [4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020. 2
- [5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 12
- [6] Nadine Chang, Zhiding Yu, Yu-Xiong Wang, Anima Anandkumar, Sanja Fidler, and Jose M Alvarez. Image-level or object-level? a tale of two resampling strategies for long-tailed detection. *ICML*, 2021. 2
- [7] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object detection by points. In *CVPR*, 2021. 2
- [8] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *arXiv:2102.01066*, 2021. 11, 14
- [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. 1, 2, 4, 5
- [10] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*, 2021. 3, 8
- [11] Bowen Dong, Zitong Huang, Yuelin Guo, Qilong Wang, Zhenxing Niu, and Wangmeng Zuo. Boosting weakly supervised object detection via learning bounding box adjusters. In *ICCV*, 2021. 2
- [12] Shijie Fang, Yuhang Cao, Xinjiang Wang, Kai Chen, Dahua Lin, and Wayne Zhang. Wssod: A new pipeline for weakly- and semi-supervised object detection. *arXiv:2105.11293*, 2021. 1, 2
- [13] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. In *ICCV*, 2021. 2
- [14] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 4, 7, 12, 14
- [15] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv:2104.13921*, 2021. 2, 3, 4, 5, 6, 7, 11, 12, 14
- [16] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 2, 4, 7, 11
- [17] Jianhua Han, Minzhe Niu, Zewei Du, Longhui Wei, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Joint coco and lvis workshop at eccv 2020: Lvis challenge track technical report: Asynchronous semi-supervised learning for large vocabulary instance segmentation. 2020. 5, 7
- [18] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3, 4, 7, 14
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 7
- [20] Zeyi Huang, Yang Zou, Vijayakumar Bhagavatula, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *NeurIPS*, 2020. 1, 2
- [21] Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. 7, 8
- [22] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv:1612.03651*, 2016. 7, 8
- [23] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr—modulated detection for end-to-end multi-modal understanding. *ICCV*, 2021. 3
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 12
- [25] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020. 1, 2, 6
- [26] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly supervised object detection with segmentation collaboration. In *ICCV*, 2019. 2
- [27] 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. 2
- [28] Yan Li, Junge Zhang, Kaiqi Huang, and Jianguo Zhang. Mixed supervised object detection with robust objectness transfer. *TPAMI*, 2018. 2
- [29] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *AAAI*, 2019. 2
- [30] 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. 1
- [31] Yan Liu, Zhijie Zhang, Li Niu, Junjie Chen, and Liqing Zhang. Mixed supervised object detection by transferring mask prior and semantic similarity. In *NeurIPS*, 2021. 2
- [32] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *ICLR*, 2021. 1
- [33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 6, 7, 12
- [34] Tai-Yu Pan, Cheng Zhang, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. On model calibration for long-tailed object detection and instance segmentation. *NeurIPS*, 2021. 2
- [35] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2

- [36] Pedro O Pinheiro and Ronan Collobert. Weakly supervised semantic segmentation with convolutional networks. In *CVPR*, 2015. 2
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021. 2, 3, 4, 6, 7, 8, 11, 12
- [38] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI*, 2020. 2, 6, 11
- [39] Vignesh Ramanathan, Rui Wang, and Dhruv Mahajan. Dlwl: Improving detection for lowshot classes with weakly labelled data. In *CVPR*, 2020. 2, 4, 5, 13
- [40] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 2, 4, 5, 13
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015. 3, 11
- [42] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G Schwing, and Jan Kautz. Ufo<sup>2</sup>: A unified framework towards omni-supervised object detection. In *ECCV*, 2020. 2
- [43] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihai Zelnik-Manor. Imagenet-21k pretraining for the masses. In *NeurIPS*, 2021. 4, 12
- [44] 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 *ICCV*, 2019. 1, 2, 6
- [45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 3, 4, 5
- [46] Yunhang Shen, Rongrong Ji, Yan Wang, Zhiwei Chen, Feng Zheng, Feiyue Huang, and Yunsheng Wu. Enabling deep residual networks for weakly supervised object detection. In *ECCV*, 2020. 2
- [47] Yunhang Shen, Rongrong Ji, Yan Wang, Yongjian Wu, and Liujuan Cao. Cyclic guidance for weakly supervised joint detection and segmentation. In *CVPR*, 2019. 2
- [48] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv:2005.04757*, 2020. 5
- [49] 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. 2
- [50] 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. 2
- [51] Jingru Tan, Gang Zhang, Hanming Deng, Changbao Wang, Lewei Lu, Quanquan Li, and Jifeng Dai. 1st place solution of lvis challenge 2020: A good box is not a guarantee of a good mask. *arXiv:2009.01559*, 2020. 7
- [52] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020. 12
- [53] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *TPAMI*, 2018. 1, 2
- [54] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 2
- [55] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *CVPR*, 2018. 2
- [56] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013. 2
- [57] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-milcontinuation multiple instance learning for weakly supervised object detection. In *CVPR*, 2019. 2
- [58] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *CVPR*, 2021. 2, 7
- [59] 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 Multimedia*, 2020. 2
- [60] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 4, 12
- [61] 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. *ICCV*, 2021. 1
- [62] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv:1702.08740*, 2017. 2
- [63] Hao Yang, Hao Wu, and Hao Chen. Detecting 11k classes: Large scale object detection without fine-grained bounding boxes. In *ICCV*, 2019. 2
- [64] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, 2019. 2
- [65] Keren Ye, Mingda Zhang, Adriana Kovashka, Wei Li, Dafen Qin, and Jesse Berent. Cap2det: Learning to amplify weak caption supervision for object detection. In *ICCV*, 2019. 3, 6
- [66] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 2, 3, 6, 8, 11
- [67] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. Mosaicos: A simple and effective use of object-centric images for long-tailed object detection. *ICCV*, 2021. 2, 4, 7, 8, 12, 14
- [68] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021. 2
- [69] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *ECCV*. Springer, 2020. 2
- [70] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv:2103.07461*, 2021. 2, 4, 6, 7, 8, 12
- [71] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. *arXiv:2102.13086*, 2021. 5
- [72] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *CVPR*, 2020. 6
- [73] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2021. 8

	$AR_r$ 50@100	$AR_r$ 50@300	$AR_r$ 50@1k	AR50@1k
LVIS-all	63.3	76.3	79.7	80.9
LVIS-base	62.2	76.2	78.5	81.0

**(a) Proposal networks trained with (top) and without (bottom) rare classes.** We report recalls on rare classes and all classes at IoU threshold 0.5 with different number of proposals. Proposal networks trained *without* rare classes can generalize to rare classes in testing.

	$AR_{half-1st}$ 50@1k	$AR_{half-2nd}$ 50@1k
LVIS-half-1st	80.8	69.6
LVIS-half-2nd	62.9	82.2

**(b) Proposal networks trained on half of the LVIS classes.** We report recalls at IoU threshold 0.5 on the other half classes. Proposal networks produce non-trivial recalls on novel classes.

**Table 11. Proposal network generalization ability evaluation.** (a): Generalize from 886 LVIS base classes to the 317 rare classes; (b): Generalize from uniformly sampled half LVIS classes (601/602 classes) to the other half.

## A. Region proposal quality

In this section, we show the region proposal network trained on LVIS [16] is satisfactory and can generalize to new classes by default. We experiment under our strong baseline in § 4.1. Table 11a shows the proposal recalls with or without rare classes in training. First, we observe the recall gaps between the two models on rare classes are small (79.7 vs. 78.5); second, the gaps between rare classes and all classes are small (79.7 vs. 80.9); third, the absolute recall is relatively high (~80%, note recall at IoU threshold 0.5 can be translated into oracle mAP-pool [8] given perfect classifier and regressor). All observations indicate the proposals can generalize to new classes even though they are supervised to background during training. These results are consistent with ViLD [15].

We in addition evaluate a more strict setting, where we uniformly split LVIS classes into two halves. I.e., we use classes ID 1, 3, 5, … as the first half, and the rest as the second half. These two subsets have completely different definitions of “objects”. We then train a proposal network on each of them, and evaluate on both subsets. As shown in Table 11b, the proposal networks give non-trivial recalls at the complementary other half. This again supports proposal networks trained on a diverse vocabulary learned a general concept of objects.

## B. Open-vocabulary COCO benchmark details

Open-vocabulary COCO is proposed by Bansal et al. [2]. They manually select 48 classes from the 80 COCO classes as base classes, and 17 classes as novel classes. The training set is the same as the full COCO, but only images containing at least one base class are used. During testing, we report results under the “generalized zero-shot detection” setting [2], where all COCO validation images are used.

	Supervision	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>
Box-Supervised	-	30.2	16.4
Detic w. CC	Image label	<b>31.0</b>	19.8
Detic w. CC	Caption	30.4	17.4
Detic w. CC	Both	<b>31.0</b>	<b>21.3</b>
		mAP50 <sub>all</sub> <sup>box</sup>	mAP50 <sub>novel</sub> <sup>box</sup>
Box-Supervised	-	39.3	1.3
Detic w. COCO-cap.	Image label	44.7	24.1
Detic w. COCO-cap.	Caption	43.8	21.0
Detic w. COCO-cap.	Both	<b>45.0</b>	<b>27.8</b>

**Table 12. Direct caption supervision.** Top: Open-vocabulary LVIS with Conceptual Caption as weakly-labeled data; Bottom block: Open-vocabulary COCO with COCO-caption as weakly-labeled data. Directly using caption embeddings as a classifier is helpful on both benchmarks; the improvements are complementary to using image labels.

We strictly follow the literatures [2, 38, 66] to use Faster-RCNN [41] with ResNet50-C4 backbone and the 1× training schedule (90k iterations). We use horizontal flip as the only data augmentation in training and keep the input resolution fixed to 800 × 1333 in both training and testing. We use SGD optimizer with a learning rate 0.02 (dropped by 10× at 60k and 80k iteration) and batch size 16. The evaluation metric on open-vocabulary COCO is box mAP at IoU threshold 0.5. Our reproduced baseline matches OVR-CNN [66]. Our model is finetuned on the baseline model with another 1× schedule. We sample detection data and image-supervised data in a 1 : 1 ratio.

## C. Direct captions supervision

As we are using a language model CLIP [37] as the classifier, our framework can seamlessly incorporate the free-form caption text as image-supervision. Using the notations in § 3.2, here  $\mathcal{D}^{cls} = \{(\mathbf{I}, t)_i\}$  where  $t$  is a free-form text. In our open-vocabulary detection formulation, text  $t$  can naturally be converted to an embedding by the CLIP [37] language encoder  $\mathcal{L}$ :  $w = \mathcal{L}(t)$ . Given a minibatch of  $B$  samples  $\{(\mathbf{I}, t)_i\}_{i=1}^B$ , we compose a dynamic classification layer by stacking all caption features within the batch  $\widetilde{\mathbf{W}} = \mathcal{L}(\{t_i\}_{i=1}^B)$ . For the  $i$ -th image in the minibatch, its “classification” label is the  $i$ -th text, and other texts are negative samples. We use the injected whole image box to extract ROI feature  $\mathbf{f}'_i$  for image  $i$ . We use the same binary cross entropy loss as classifying image labels:

$$L_{cap} = \sum_{i=1}^B BCE(\widetilde{\mathbf{W}}\mathbf{f}'_i, i)$$

We do not back-propagate into the language encoder.

We evaluate the effectiveness of the caption loss in Table 12 on both open-vocabulary LVIS and COCO. We com-

	mAP <sup>box</sup>	mAP <sup>box</sup> <sub>r</sub>	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>r</sub>	T
D2 baseline [60]	22.9	11.3	22.4	11.6	12h
+Class-agnostic box&mask	22.3	10.1	21.2	10.1	12h
+Federated loss [70]	27.0	20.2	24.6	18.2	12h
+CenterNet2 [70]	30.7	22.9	26.8	19.4	13h
+LSJ $640 \times 640$ , 4× sched. [14]	31.0	21.6	27.2	20.1	17h
+CLIP classifier [37]	31.5	24.2	28	22.5	17h
+Adam optimizer, lr $2e-4$ [24]	30.4	23.6	26.9	21.4	17h
+IN-21k pretrain [43]*	35.3	28.2	31.5	25.6	17h
+Input size $896 \times 896$	37.1	29.5	33.2	26.9	25h
+Swin-B backbone [33]	45.4	39.9	40.7	35.9	43h
*Remove rare class ann.[15]	33.8	17.6	30.2	16.4	17h

**Table 13. LVIS baseline evolution.** First row: the configuration from the detectron2 model zoo. The following rows change components one by one. Last row: removing rare class annotation from the “+IN-21k pretrain\*” row. The two gray-filled rows are the baselines in our main paper, for full LVIS and open-vocabulary LVIS, respectively. We show the rough wall-clock training time ( $T$ ) on our machine with 8 V100 GPUs in the last column.

pare individually applying the max-size loss for image labels and the caption loss, and applying both of them. Both image labels and captions can improve both overall mAP and novel class mAP. Combining both losses gives a more significant improvement. Our open-vocabulary COCO results in Table 3 uses both the max-size loss and the caption loss.

## D. LVIS baseline details

We first describe the standard LVIS baseline from the detectron2 model zoo<sup>4</sup>, which is exactly the baseline used in MosaicOS [67]. This baseline uses ResNet-50 FPN backbone and a  $2\times$  training schedule (180k iterations with batch-size 16)<sup>5</sup>. Data augmentation includes horizontal flip and random resize short side [ $640, 800$ ], long side  $< 1333$ . The baseline uses SGD optimizer with a learning rate 0.02 (dropped by 10× at 120k and 160k iteration). The bounding box regression head and the mask head are class-specific.

Table 13 shows the roadmap from the detectron2 baseline to our baseline (§ 4.1). First, we prepare the model for new classes by making the box and mask heads class-agnostic. This slightly hurts performance. We then use Federated loss [70] and upgrade the detector to CenterNet2 [70] (i.e., replacing RPN with CenterNet and multiplying proposal score to classification score). Both modifications improve mAP and mAP<sub>r</sub> significantly, and CenterNet2 slightly increases the training time.

Next, we use the EfficientDet [14, 52] style large-scale jittering and train a longer schedule (4×). To balance the

<sup>4</sup>[https://github.com/facebookresearch/detectron2/blob/main/configs/LVISv1-InstanceSegmentation/mask\\_rcnn\\_R\\_50\\_FPN\\_1x.yaml](https://github.com/facebookresearch/detectron2/blob/main/configs/LVISv1-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml)

<sup>5</sup>We are aware different projects use different notations of a 1× schedule. In this paper we always refer 1× schedule to  $16 \times 90k$  images

	Ratio	Size	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>
Bos-Supervised	1: 0	-	30.2	16.4
Detic w. IN-L	1: 1	640	30.9	23.3
Detic w. IN-L	1: 1	320	32.0	24.0
Detic w. IN-L	1: 4	640	31.1	23.5
Detic w. IN-L	1: 4	320	<b>32.4</b>	<b>24.9</b>
Detic w. CC	1: 1	640	30.8	21.6
Detic w. CC	1: 1	320	30.8	21.5
Detic w. CC	1: 4	640	30.7	21.0
Detic w. CC	1: 4	320	<b>31.1</b>	<b>21.8</b>

**Table 14. Ablations of the resolution change.** We report mask mAP on the open-vocabulary LVIS following the setting of Table 1. Top: ImageNet as the image-labeled data. Bottom: CC as the image-labeled data.

training time, we also reduce the training image size to  $640 \times 640$  (the testing size is unchanged at  $800 \times 1333$ ) and increase batch-size to 64 (with the learning rate scaled up to 0.08). The resulting augmentation and schedule is slightly better than the default multi-scale training, with 30% more training time. A longer schedule is beneficial when using more data, and can be improved by larger resolution.

Next, we switch in the CLIP classifier [37]. We follow ViLD [15] to L2 normalize the embedding and RoI feature before dot-product. Note CenterNet2 uses a cascade classifier [5]. We use CLIP for all of them. Using CLIP classifier improves rare class mAP.

Finally, we use an ImageNet-21k pretrained ResNet-50 model from Ridnik *et al.* [43]. We remark the ImageNet-21k pretrained model requires using Adam optimizer (with learning rate  $2e-4$ ). Combing all the improvements results in 35.3 mAP<sup>box</sup> and 31.5 mAP<sup>mask</sup>, and trains in a favorable time (17h on 8 V100 GPUs). We use this model as our baseline in the main paper.

Increasing the training resolution or using a larger backbone [33] can further increase performance significantly, at a cost of longer training time. We use the large models only when compared to the state-of-the-art models.

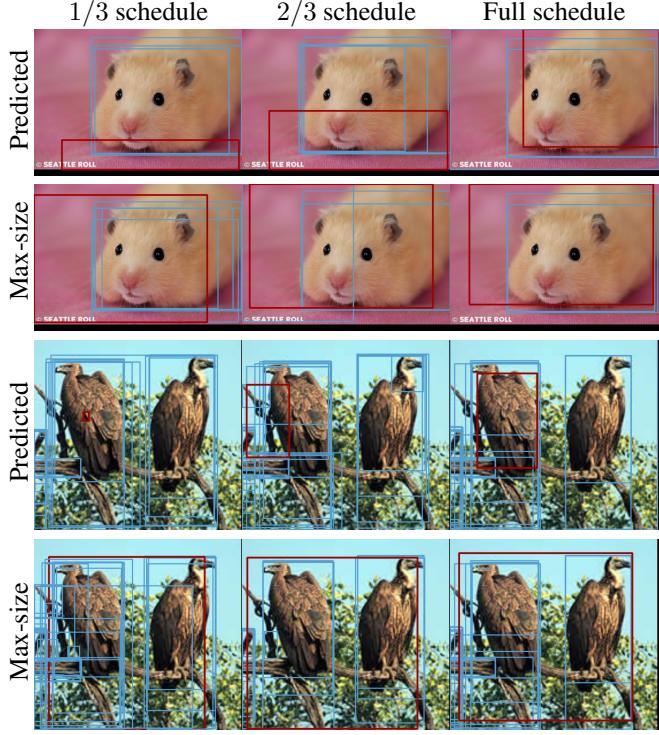
## E. Resolution change for classification data

Table 14 ablates the resolution change in § 4.1. Using a smaller input resolution improves  $\sim 1$  point for both mAP and mAP<sub>novel</sub> with ImageNet, but does not impact much with CC. Using more batches for the weak datasets is slightly better than a 1 : 1 ratio.

## F. WSOD losses implementation details

Following the notations in § 3.2, we implement the prediction-based WSOD losses as below:

**WSDDN** [3] learns a soft weight on the proposals to weight-



**Figure 5. Visualization of the assigned proposals.** We show all proposals with score  $> 0.5$  in blue and the assigned proposal in red. We compare the predicted loss with our max-size loss across different training iterations. Max-size loss in most case covers the target object, and is consistent during training.

sum the proposal classification scores into a single image classification score:

$$L_{WSDDN} = BCE\left(\sum_j (\text{softmax}(\mathbf{W}' \mathbf{F})_j * \mathbf{S}_j), c\right)$$

where  $\mathbf{W}'$  is a learnable network parameter.

**Predicted** [40] selects the proposal with the max predicted score on class  $c$ :

$$L_{\text{Predicted}} = BCE(\mathbf{S}_j, c), j = \text{argmax}_j \mathbf{S}_{jc}$$

**DLWL\*** [39] first runs a clustering algorithm with IoU threshold 0.5. Let  $\mathcal{J}$  be the set of peaks of each cluster (i.e., the proposal within the cluster and has the max predicted score on class  $c$ ). We then select the top  $N_c = 3$  peaks with the highest prediction scores on class  $c$ .

$$L_{\text{DLWL}^*} = \frac{1}{N_c} \sum_{t=1}^{N_c} BCE(\mathbf{S}_{j_t}, c),$$

$$j_t = \text{argmax}_{j \in \mathcal{J}, j \neq \{j_1, \dots, j_{t-1}\}} \mathbf{S}_{jc}$$

The original DLWL [39] in addition upgrades  $\mathbf{S}$  using an IoU-based assignment matrix from self-training and bootstrapping (See their Section 3.2). In our implementation, we did not include this part, as our goal is to only compare the training losses.

	Dataset	Backbone	mAP <sup>mask</sup>	mAP <sup>mask</sup> <sub>novel</sub>
Box-Supervised	LVIS-base	Res50	30.2	16.4
Predicted			31.2	20.4
Max-size			32.4 (+1.2)	24.6 (+4.2)
Box-Supervised	LVIS-base	SwinB	38.4	21.9
Predicted			40.0	31.7
Max-size			40.7 (+0.7)	33.8 (+2.1)
Box-Supervised	LVIS-all	Res50	31.5	25.6
Predicted			32.5	28.4
Max-size			33.2 (+0.7)	29.7 (+1.3)
Box-Supervised	LVIS-all	SwinB	40.7	35.9
Predicted			40.6	39.8
Max-size			41.3 (+0.7)	40.9 (+1.1)

**(a) Predicted loss and max-size loss with different prediction qualities.** We show the mask mAP of the box-supervised baseline, Predicted loss [40], and our max-size loss. We show the delta between max-size loss and predicted loss in green. Improving the backbone and including rare classes in training can both narrow the gap. Max-size consistently performs better.

	Cover rate		Consistency		
	IN-L	COCO	IN-L	CC	COCO
Predicted	69.0	73.8	71.5	30.0	57.7
Max-size	92.8	80.0	87.9	73.0	62.8

**(b) Assigned proposal cover rate and consistency.** Left: ratio of assigned proposal covering the ground truth both. We evaluate on an ImageNet subset that has box ground truth and the annotated COCO training set; Right: average assigned bounding box IoU of between the final model and the half-schedule model.

**Table 15. Comparison between predicted loss and and max-size loss.** **(a):** comparison under different baselines. **(b):** comparison in customized metrics.

## G. More comparison to predicted loss

Our max-size loss performs significantly better than WSOD losses that are based on predictions as is shown in Table 1. In this section, we provide more detailed comparisons to the **Predicted** loss as a representative of prediction-based methods. A straightforward reason is the predicted loss requires a good initial prediction to guide the pseudo-label-based training. However in the open-vocabulary detection setting the initial predictions are inherently flawed. To verify this, in Table 15a, we show both improving the backbone and including rare classes in training can narrow the gap. However in the current performance regime, our max-size loss performs better.

We highlight two additional advantages of the max-size loss that may contribute to the good performance: (1) the max-size loss is a safe approximation of object regions; (2) the max-size loss is consistent during training. Figure 5 provides qualitative examples of the assigned region for the predicted loss and the max-size loss. First, we observe that while being coarse at the boundary, the max-size loss can

	mAP <sup>mask</sup>	mAP <sub>r</sub> <sup>mask</sup>
Box-Supervised [67]	22.6	12.3
MosaicOS [67]	24.5 (+1.9)	18.3 (+6.0)
Box-Supervised (Reproduced)	22.6	12.3
Detic (default classifier)	25.1 (+2.5)	18.6 (+6.3)
Box-Supervised (CLIP classifier)	22.3	14.1
Detic (CLIP classifier)	<b>24.9 (+2.6)</b>	<b>20.7 (+6.5)</b>

**Table 16. Detailed comparison to MosaicOS [67].** Top block: results quoted from MosaicOS paper; Middle block: Detic with the default random initialized and trained classifier; Bottom block: Detic with CLIP classifier.

cover the target object in most cases. Second, the assigned regions of the predicted loss are usually different across training iterations, especially in the early phase where the model predictions are unstable. On the contrary, max-size loss supervises consistent regions across training iterations.

Table 15b quantitatively evaluates these two properties. We use the ground truth box annotation in the full COCO detection dataset and a subset of ImageNet with bounding box annotation<sup>6</sup> to evaluate the cover rate. We define cover rate as the ratio of image labels whose ground-truth box has  $> 0.5$  intersection-over-area with the assigned region. We define the consistency metric as the average assigned-region IoU of the same image between the 1/2 schedule and the final schedule. Table 15b shows max-size loss is more favorable than predicted loss on these two metrics. However we highlight that these two metrics alone do not always correlate to the final performance, as the **image-box** loss is perfect on both metrics but underperforms max-size loss.

## H. ViLD baseline details

The baseline in ViLD [15] is very different from detectron2. They use MaskRCNN detector [18] with Res50-FPN backbone, but trains the network from scratch without ImageNet pretraining. They use large-scale jittering [14] with input resolution  $1024 \times 1024$  and train a  $32 \times$  schedule. The optimizer is SGD with batch size 256 and learning rate 0.32. We first reproduce their baselines (both the oracle detector and ViLD-text) under the same setting. We observe half of their schedule ( $16 \times$ ) is sufficient to closely match their numbers. The half training schedule takes 4 days on 4 nodes (each with 8 V100 GPUs). We then finetune another  $16 \times$  schedule using ImageNet data with our max-size loss.

## I. Details of comparison to MosaicOS

We compare to MosaicOS [67] by strictly following their baseline setup. The detailed hyper-parameters follow the detectron2 baseline as described in Appendix D. We finetune

<sup>6</sup><https://image-net.org/download-bboxes.php>. 213K of the 1.2M IN-L images have bounding box annotations.

	mAP <sup>mask</sup>	mAP <sub>IN-L</sub> <sup>mask</sup>	mAP <sub>non-IN-L</sub> <sup>mask</sup>
Box-Supervised	30.2	30.6	27.6
Max-size	32.4	33.5	28.1
	mAP <sup>mask</sup>	mAP <sub>CC</sub> <sup>mask</sup>	mAP <sub>non-CC</sub> <sup>mask</sup>
Box-Supervised	30.2	30.1	29.5
Max-size	30.9	31.7	28.6

**Table 17. mAP breakdown into classes with and without image labels.** Top: Detic trained on ImageNet. Bottom: Detic trained on CC. Most of the improvements are from classes with image-level labels. On ImageNet Detic also improves classes without image labels thanks to the CLIP classifier.

Datasets	mAP <sup>box</sup>	mAP <sup>box</sup> <sub>novel</sub>	mAP <sup>Fixed</sup>	mAP <sup>Fixed</sup> <sub>novel</sub>
Box-Supervised	30.2	16.4	31.2	18.2
Detic	32.4 (+2.2)	24.9 (+8.5)	33.4 (+2.3)	26.7 (+8.5)

**Table 18. mAP<sup>Fixed</sup> evaluation.** Middle: the original box mAP metric used in the main paper. Right: the new box mAP<sup>Fix</sup> metric. Our improvements are consistent under the new metric.

on the Box-supervised model with an additional  $2 \times$  schedule with Adam optimizer. Table 16 shows our re-trained baseline exactly matches their reported results from the paper. Our method is developed based on the CLIP classifier, and we also report our baseline with CLIP. The baseline has slightly lower mAP and higher mAP<sub>r</sub>. Our relative improvements over the baseline are slightly higher than MosaicOS [67]. We highlight our training framework is simpler and we use less additional training data (Google-searched images).

## J. Improvements breakdown to classes

Table 17 shows mAP breakdown into classes with and without image labels for both the Box-Supervised baseline and Detic. As expected, most of the improvements are from classes with image-level labels. On ImageNet, Detic also improves classes without image labels thanks to the CLIP classifier which leverages inter-class relations.

## K. mAP<sup>Fixed</sup> evaluation

Table 18 compares our improvements under the new mAP<sup>fix</sup> proposed in Dave *et al.* [8]. Our improvements are consistent under the new metric.

## L. Image Attributions

License for the images from OpenImages in Figure 4:

- “Oyster”: Photo by The Local People Photo Archive (CC BY 2.0)
- “Cheetah”: Photo by Michael Gil (CC BY 2.0)
- “Harbor seal”: Photo by Alden Chadwick (CC BY 2.0)
- “Dinosaur”: Photo by Paxson Woelber (CC BY 2.0)