	$AR_r50@100$	$AR_r50@300$	$AR_r50@1k$	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.

	$AR_{\text{half-1st}}50@1k$	$AR_{\text{half-2nd}}50@1k$
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 8: Proposal network generalization ability evaluation. (a): Generalize from 866 LVIS base classes to the 337 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 [18] is satisfactory and generalizes well to new classes by default. We experiment under our strong baseline in § 5.1. Table 8a 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 ($\sim 80\%$, 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 have good generalization abilities to new classes even though they are supervised to background during training. We consider the proposal generalization is currently not the performance bottleneck in open-vocabulary detection. This especially the case as modern detectors use an over-sufficient number of proposals in testing (1K proposals for < 20 objects per image). Our observations are consistent with ViLD [17].

We in addition evaluate a more strict setting, where we uniformly split LVIS classes into two halves. I.e., we use classes ID $1, 3, 5, \cdots$ 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 8b, the proposal networks give non-trivial recalls at the complementary other half (69.6% over 82.2% percent of the full generalizability). This again supports proposal networks trained on a diverse vocabulary learned a general concept of objects.

B Direct captions supervision

As we are using a language model CLIP [42] as the classifier, our framework can seamlessly incorporate the free-form caption text as image-supervision. Using

	Supervision	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{novel}}$
Box-Supervised	-	30.2	16.4
Detic w. CC	Image label	**31.0**	19.8
Detic w. CC	Caption	30.4	17.4
Detic w. CC	Both	**31.0**	**21.3**
		$\text{mAP50}^{\text{box}}_{\text{all}}$	$\text{mAP50}^{\text{box}}_{\text{novel}}$
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	**45.0**	**27.8**

Table 9: 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 Detic.

the notations in § 4, here $\mathcal{D}^{\text{cls}} = \{(\mathbf{I}, t)_i\}$ where t is a free-form text. In our open-vocabulary detection formulation, text t can natrually be converted to an embedding by the CLIP [42] 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 9 on both open-vocabulary LVIS and COCO (see dataset details in Appendix H). We compare 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.

C LVIS baseline details

We first describe the standard LVIS baseline from the detectron2 model zoo[3]. This baseline uses ResNet-50 FPN backbone and a 2× training schedule (180k

[3] https://github.com/facebookresearch/detectron2/blob/main/configs/LVISv1-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml

	mAP^{box}	$\text{mAP}^{\text{box}}_{\text{r}}$	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{r}}$	T
D2 baseline [66]	22.9	11.3	22.4	11.6	12h
+Class-agnostic box&mask	22.3	10.1	21.2	10.1	12h
+Federated loss [76]	27.0	20.2	24.6	18.2	12h
+CenterNet2 [76]	30.7	22.9	26.8	19.4	13h
+LSJ 640×640, 4× sched. [15]	31.0	21.6	27.2	20.1	17h
+CLIP classifier [42]	31.5	24.2	28	22.5	17h
+Adam optimizer, lr2e-4 [26]	30.4	23.6	26.9	21.4	17h
+IN-21k pretrain [48]*	35.3	28.2	31.5	25.6	17h
+Input size 896×896	37.1	29.5	33.2	26.9	25h
+Swin-B backbone [37]	45.4	39.9	40.7	35.9	43h
*Remove rare class ann.[17]	33.8	17.6	30.2	16.4	17h

Table 10: LVIS baseline evolution. First row: the configuration from the detectron2 model zoo. The following rows change components one by one. Last row: removing rare classes 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 rough wall-clock training times (T) on our machine with 8 V100 GPUs in the last column.

iterations with batch-size 16)[4]. 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 10 shows the roadmap from the detectron2 baseline to our baseline (§ 5.1). 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 [76] and upgrade the detector to CenterNet2 [76] (i.e., replacing RPN with CenterNet and multiplying proposal score to classification score). Both modifications improve mAP and mAP_{r}significantly, and CenterNet2 slightly increases the training time.

Next, we use the EfficientDet [15, 58] style large-scale jittering and train a longer schedule (4×). To balance the training time, we also reduce the training image size to 640×640 (the testing size is unchanged at 800×1333) 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 [42]. We follow ViLD [17] 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.* [48]. We remark the ImageNet-21k pretrained model requires using Adam optimizer (with learning rate 2e-4). Combing all the improvements results in

[4] 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^{mask}	$\text{mAP}^{\text{mask}}_{\text{novel}}$
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	**32.4**	**24.9**
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	**31.1**	**21.8**

Table 11: 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.

35.3 mAP^{box} and 31.5 mAP^{mask}, 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 [37] 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.

D Resolution change for classification data

Table 11 ablates the resolution change in § 5.1. Using a smaller input resolution improves $\sim$1 point for both mAP and $\text{mAP}_{\text{novel}}$ with ImageNet, but does not impact much with CC. Using more batches for the weak datasets is slightly better than a $1:1$ ratio.

E Prediction-based losses implementation details

Following the notations in § 4, we implement the prediction-based weakly-supervised detection losses as below:

WSDDN [3] learns a soft weight on the proposals to weight-sum the proposal classification scores into a single image classification score:

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

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

Predicted [45] 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* [44] 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.

$$L_{\text{DLWL}^*} = \frac{1}{N_c}\sum_{t=1}^{N_c} BCE(\mathbf{S}_{j_t}, c),$$
$$j_t = \text{argmax}_{j\in\mathcal{J}, j\neq\{j_1,\ldots,j_{t-1}\}}\mathbf{S}_{jc}$$

The original DLWL [44] 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.

F More comparison between prediction-based and non-prediction-based methods

Our non-prediction-based losses perform significantly better than prediction-based losses as is shown in Table 1. In this section, we take the max-size loss and the predicted-loss as the representitives and conduct more detailed comparisons between them. A straightforward reason is that 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 12a, 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 4 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 *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 12b 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 [5] 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 12b 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.

[5] https://image-net.org/download-bboxes.php. 213K of the 1.2M IN-L images have bounding box annotations.

	Dataset	Backbone	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{novel}}$
Box-Supervised			30.2	16.4
Predicted	LVIS-base	Res50	31.2	20.4
Max-size			32.4 (+1.2)	24.6 (+4.2)
Box-Supervised			38.4	21.9
Predicted	LVIS-base	SwinB	40.0	31.7
Max-size			40.7 (+0.7)	33.8 (+2.1)
Box-Supervised			31.5	25.6
Predicted	LVIS-all	Res50	32.5	28.4
Max-size			33.2 (+0.7)	29.7 (+1.3)
Box-Supervised			40.7	35.9
Predicted	LVIS-all	SwinB	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 [45], 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 12: Comparison between predicted loss and and max-size loss. (a): comparison under different baselines. **(b)**: comparison in customized metrics.

G ViLD baseline details

The baseline in ViLD [17] is very different from detectron2. They use MaskRCNN detector [20] with Res50-FPN backbone, but trains the network from scratch without ImageNet pretraining. They use large-scale jittering [15] with input resolution 1024×1024 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.

H 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.

	$\text{mAP50}^{\text{box}}_{\text{all}}$	$\text{mAP50}^{\text{box}}_{\text{novel}}$
Box-Supervised (base cls)	39.3	1.3
Self-training [54]	39.5	1.8
WSDDN [3]	39.9	5.9
DLWL* [44]	42.9	19.6
Predicted [45]	41.9	18.7
Detic (Max-object-score)	43.3	20.4
Detic (Image-box)	43.4	21.0
Detic (Max-size)	**44.7**	**24.1**
Box-Supervised (all cls)	54.9	60.0

Table 13: Different ways to use image supervision on open-vocabulary COCO. The models are trained using the OVR-CNN [72] recipe with ResNet50-C4 [2] backbone. We follow setups in Table 1. The observations are consistent with LVIS.

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.

We strictly follow the literatures [2, 43, 72] to use FasterRCNN [46] 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 [72]. 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.

Table 13 repeats the experiments in Table 1 on open-vocabulary COCO. The observations are consistent: our proposed non-prediction-based methods outperform existing prediction-based counterparts, and the max-size loss performs the best among our variants.

I Compare to MosaicOS [73]

MosaicOS [73] first uses image-level annotations to improve LVIS detectors. We compare to MosaicOS [73] by strictly following their baseline setup (without any improvements in § 5.1). The detailed hyper-parameters follow the detectron2 baseline as described in Appendix C. We finetune on the Box-supervised model with an additional 2× schedule with Adam optimizer. Table 14 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_r. MosaicOS uses IN-L and additional web-search images as image-supervised data. Detic outperforms MosaicOS [73] in mAP and mAP_r, without using their multi-stage

	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{r}}$
Box-Supervised [73]	22.6	12.3
MosaicOS [73]	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)	**24.9 (+2.6)**	**20.7 (+6.5)**

Table 14: Standard LVIS compared to MosiacOS [73]. Top block: results quoted from MosiacOS paper; Middle block: Detic with the default random intialized and trained classifier; Bottom block: Detic with CLIP classifier.

	mAP^{box}	$\text{mAP}^{\text{box}}_{\text{r}}$	$\text{mAP}^{\text{box}}_{\text{c}}$	$\text{mAP}^{\text{box}}_{\text{f}}$
Box-Supervised	31.7	21.4	30.7	**37.5**
Detic	**32.5**	**26.2**	**31.3**	36.6

Table 15: Detic applied to Deformable-DETR [79]. We report Box mAP on full LVIS. Our method improves Deformable-DETR.

training and mosaic augmentation. Our relative improvements over the baseline are slightly higher than MosiacOS [73]. We highlight our training framework is simpler and we use less additional training data (Google-searched images).

J Generalization to Deformable-DETR.

We apply Detic to the recent Transformer based Deformable-DETR [79] to study its generalization. We use their default training recipe, Federated Loss [76] and train for a 4× schedule (∼48 LVIS epochs). We apply the image supervision to the query from the encoder with the max predicted size. Table 15 shows that Detic improves over the baseline (+0.8 mAP and +4.8 mAP_{r}) and generalizes to Transformer based detectors.

	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{IN-L}}$	$\text{mAP}^{\text{mask}}_{\text{non-IN-L}}$
Box-Supervised	30.2	30.6	27.6
Max-size	32.4	33.5	28.1
	mAP^{mask}	$\text{mAP}^{\text{mask}}_{\text{CC}}$	$\text{mAP}^{\text{mask}}_{\text{non-CC}}$
Box-Supervised	30.2	30.1	29.5
Max-size	30.9	31.7	28.6

Table 16: 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^{box}	$\text{mAP}^{\text{box}}_{\text{novel}}$	$\text{mAP}^{\text{Fixed}}$	$\text{mAP}^{\text{Fixed}}_{\text{novel}}$
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 17: $\text{mAP}^{\text{Fixed}}$ evaluation. Middle: the original box mAP metric used in the main paper. Right: the new box mAP^{Fix} metric. Our improvements are consistent under the new metric.

K Improvements breakdown to classes

Table 16 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.

L $\text{mAP}^{\text{Fixed}}$ evaluation

Table 17 compares our improvements under the new mAP^{fix} proposed in Dave *et al.* [8]. Our improvements are consistent under the new metric.

M Image Attributions

License for the images from OpenImages in Figure 5:

- "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)