Expanding Low-Density Latent Regions for Open-Set Object Detection Supplementary Material

A. More Experimental Details

A.1. Datasets

In this section, we introduce more details about the dataset construction.

PASCAL VOC [2]. We use VOC07 `train` and VOC12 `trainval` splits for the training, and VOC07 `test` split to evaluate the close-set performance. We take VOC07 `val` as the validation set.

VOC-COCO-$\mathbf{T}_1$. We divide 80 COCO classes into four groups (20 classes per group) by their semantics [7]: **(1)** VOC classes. **(2)** Outdoor, Accessories, Appliance, Truck. **(3)** Sports, Food. **(4)** Electronic, Indoor, Kitchen, Furniture. We construct VOC-COCO-$\{20, 40, 60\}$ with n=5000 VOC testing images and $\{n, 2n, 3n\}$ COCO images **containing** $\{20, 40, 60\}$ non-VOC classes with semantic shifts, respectively. Note that we only ensure each COCO image contains objects of corresponding open-set classes, which means objects of VOC classes will also appear in these images. This setting is more similar to real-world scenarios where detectors need to carefully identify unknown objects and do not classify known objects into the unknown class.

VOC-COCO-$\mathbf{T}_2$. We gradually increase the Wilderness Ratio to build four dataset with n=5000 VOC testing images and $\{0.5n, n, 2n, 4n\}$ COCO images **disjointing** with VOC classes. Compared with the setting $\mathbf{T}_1$, $\mathbf{T}_2$ aims to evaluate the model under a higher wilderness, where large amounts of testing instances are not seen in the training.

Comparisons with existing benchmarks. [1] proposed the first OSOD benchmark. They also use the data in VOC for close-set training, and both VOC and COCO for open-set testing. In the testing phase, they just vary the number of open-set images sampled from COCO, while ignoring the number of open-set categories. [7] proposed an open world object detection benchmark. They divide the open-set testing set into several groups by category. However, the wilderness ratio of each group is limited, and such data partitioning cannot reflect the real performance of detectors under extreme open-set conditions. In contrast, our proposed benchmark considers both the number of open-set classes (VOC-COCO-T_1) and images (VOC-COCO-T_2).

On the other hand, some works on open-set panoptic segmentation [6] divide a single dataset into close-set and open-set. If a image contains both close-set and open-set instances, they just remove the annotations of open-set instances. Differently, we strictly follows the definition in OSR [13] that unknown instances should not appear in training. To acquire enough open-set examples, we take both VOC and COCO from cross-dataset evaluation, which is a common practice in OSR [9, 14, 16].

A.2. Implementation Details

Training schedule. Inspired by [15] that a good close-set classifier benefits OSR, we train all models with the 3× schedule (*i.e.*, 36 epochs). Besides, we enable UPL after several warmup iterations (*e.g.*, 100 iterations) to make sure the model produce valid probabilities.

Open-RetinaNet. We change some hyper-parameters for Open-RetinaNet. In OpenDet, we take object proposals as examples and apply CFL to proposal-wise embeddings, which are equivalent to the anchor boxes in RetinaNet. Therefore, we optimize Instance Contrastive Loss $\mathcal{L}_{IC}$ with pixel-wise features of each anchor box. Since the number of anchor box is much larger than the proposals in OpenDet, we enlarge the memory size Q=1024, sampling size q=64, and loss weight to 0.2 in CFL. Similar, we sample 10 hard examples rather than 3 in UPL.

A.3. Evaluation Metrics

Firstly, we give a detailed formulation of the Wilderness Impact [1], which is defined as:

$$\begin{aligned} WI &= \frac{P_{\mathcal{K}}}{P_{\mathcal{K}\cup\mathcal{U}}} - 1 \\ &= \frac{TP_{\mathcal{K}}}{TP_{\mathcal{K}} + FP_{\mathcal{K}}} \Big/ \frac{TP_{\mathcal{K}}}{TP_{\mathcal{K}} + FP_{\mathcal{K}} + FP_{\mathcal{U}}} - 1 \\ &= \frac{FP_{\mathcal{U}}}{TP_{\mathcal{K}} + FP_{\mathcal{K}}}, \end{aligned} \tag{1}$$

where $FP_{\mathcal{U}}$ means that any detections belonging to the unknown classes $C_{\mathcal{U}}$ are classified to one of known classes $C_{\mathcal{K}}$. For $\mathrm{AP}_{\mathcal{U}}$ (AP of unknown classes), we merge the annotations of all unknown classes into one class, and calculate the *class-agnostic* AP between unknown's predictions and the ground truth.

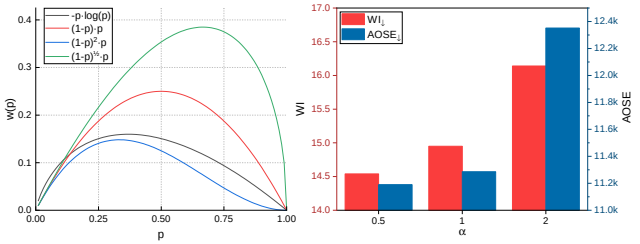


Figure A1. **Visualization of different $w(\cdot)$.**

B. Additional Main Results

Due to limited space in our main paper, we report the results on VOC-COCO-2n in Tab. A1, where OpenDet shows significant improves than other methods.

Method	WI$_\downarrow$	AOSE$_\downarrow$	mAP$_{\mathcal{K}\uparrow}$	AP$_{\mathcal{U}\uparrow}$
FR-CNN [12]	24.18	24636	70.07	0
FR-CNN* [12]	24.05	18740	69.81	0
PROSER [16]	25.74	21107	69.32	10.31
ORE [7]	23.67	20839	70.01	2.13
DS [11]	23.21	20018	69.33	4.84
OpenDet	**18.69**	**16329**	**71.44**	**14.96**

Table A1. **Comparisons with other methods on VOC-COCO-2n.** This table is an extension of Tab.2 in our main paper.

C. Additional Ablation Studies

Visual analyses of $w(\cdot)$. In Fig. A1, we plot the graph of different $w(\cdot)$. Compared with entropy: $-p\log(p)$, the proposed function $(1-p)^\alpha \cdot p$ can adjsut the curve shape by changing α. In other words, the model adjusts the weights of examples as α changes. The right of Fig. A1 reports the model's open-set performance by varying α, where smaller α reduces WI and AOSE.

Quantitative analyses of latent space. In Fig. **??** of the main paper, we give a visual analyses of latent space. Here we give a quantitative analyses of latent space in Tab. A2. Specifically, we calculate the intra-class variance and inter-class distance of latent features. Tab. A2 shows that CFL and UPL, as well as their combination reduce intra-class variance and enlarge inter-class distance. The results further confirm our conclusion in the main paper that our method can expand low-density latent regions.

More hyper-parameters in CFL. Loss weight: Tab. A3 shows that loss weight is important for $\mathcal{L}_{IC}$, where a small weight (*e.g.*, 0.01) cannot learn compact features and a large weight (*e.g.*, 1.0) hinder the generalization ability. Besides, Tab. A3 (last column) also demonstrates the effectiveness of loss decay. **Temperature:** We try different τ that used in pervious works [3, 8]. Tab. A4 indicates that τ=0.1 [8] works better than other settings.

metric	baseline	+CFL	+UPL	Ours
intra-variance	3.79	2.83	3.05	**2.47**
inter-distance	62.74	65.17	64.69	**66.31**

Table A2. **Quantitative analyses of the latent space.** We calculate the intra-class variance and inter-class distance of latent features.

γ_t	0.01	**0.1**	0.5	1.0	w/o decay
WI$_\downarrow$	16.13	14.95	12.26	9.71	15.65
mAP$_{\mathcal{K}\uparrow}$	58.90	58.75	57.47	53.36	58.43

Table A3. **Loss weight of $\mathcal{L}_{IC}$.** *w/o* decay: γ_t is a constant (*i.e.*, 0.1) instead of variable.

τ	0.07 [3]	**0.1** [8]	0.2
WI$_\downarrow$	15.48	14.95	15.50
mAP$_{\mathcal{K}\uparrow}$	57.80	58.75	58.87

Table A4. **Temperature τ in $\mathcal{L}_{IC}$.**

setting	backbone	epoch	WI$_\downarrow$	mAP$_{\mathcal{K}\uparrow}$
end-to-end	-	-	**14.95**	**58.75**
fine-tune	fixed	1	17.98	56.88
	fixed	12	17.43	56.86
	trainable	12	17.01	57.19

Table A5. **End-to-end *vs.* fine-tune in UPL. End-to-end:** we jointly optimize UPL and other modules in OpenDet. **Fine-tune:** we pretrain a model without UPL, and optimize UPL in the fine-tuning stage.

Training strategy. Some works in OSR [16] adopted a pretrain-then-finetune paradigm to train the unknown identifier. We carefully design the UPL so that OpenDet can be trained in an end-to-end manner. Tab. A5 shows that jointly optimizing UPL performs better than that of fine-tuning.

Open-RetinaNet. To further demonstrates the effectiveness of Open-RetinaNet, we report more results in Tab. A6, where Open-RetinaNet shows substantial improvements on WI, AOSE and AP$_{\mathcal{U}}$, and achieves comparable performance on mAP$_{\mathcal{K}}$.

Vision transformer as backbone. We find the detector with vision transformer, *e.g.*, Swin Transformer [10] is a stronger baseline for OSOD. As shown in Tab. A7, models with a Swin-T backbone significantly suppress their ResNet counterparts.

Speed and computation. In the training stage, OpenDet only increases 14% (1.4h *vs.* 1.2h) training time and 1.2% (2424Mb *vs.* 2395Mb) memory usage. In the testing phase, as we only add the unknown class to the classifier, OpenDet keeps similar running speed and computation with FR-CNN.

D. Comparison with ORE [7]

Implementation details. The original ORE adopted a R50-C4 FR-CNN framework, and train the model with 8 epochs. For fair comparisons, we replace the R50-C4 architecture

Method	$WI_{\downarrow}$	$AOSE_{\downarrow}$	$mAP_{\mathcal{K}\uparrow}$	$AP_{\mathcal{U}\uparrow}$
VOC:				
RetinaNet	-	-	**79.84**	-
Open-RetinaNet	-	-	79.72	-
VOC-COCO-40:				
RetinaNet	17.60	58383	**53.81**	0
Open-RetinaNet	**13.65**	**25964**	53.22	**8.23**
VOC-COCO-60:				
RetinaNet	14.20	64327	**54.68**	0
Open-RetinaNet	**11.28**	**30631**	54.25	**3.20**

Table A6. **Open-RetinaNet on more datasets.**

Method	backbone	$WI_{\downarrow}$	$AOSE_{\downarrow}$	$mAP_{\mathcal{K}\uparrow}$	$AP_{\mathcal{U}\uparrow}$
FR-CNN	ResNet-50	18.39	15118	58.45	0
	Swin-T	**15.99**	**13204**	**63.09**	0
OpenDet	ResNet-50	14.95	11286	58.75	14.93
	Swin-T	**12.51**	**9875**	**63.17**	**15.77**

Table A7. **Comparisons of different backbones**, *i.e.*, ResNet-50 [4] and Swin-T [10].

Method	train model on valset	$WI_{\downarrow}$	$AOSE_{\downarrow}$	$mAP_{\mathcal{K}\uparrow}$	$AP_{\mathcal{U}\uparrow}$
FR-CNN	×	18.39	15118	58.45	0
ORE	×	8.46	2909	53.96	9.64
	✓	16.98	12868	58.35	5.13
OpenDet	×	14.95	11286	58.75	14.93

Table A8. **Comparison with ORE [7].** The row with gray background is reported in our main paper.

with R50-FPN, and train all models with 3× schedule. Besides, as discussed in these issues[1], we report our re-implemented results when comparing with ORE in an open world object detection task (see Tab. A9).

Analysis of ORE. To learn the energy-based unknown identifier (Sec 4.3 in [7]), ORE requires an additional validation set with the annotations of unknown classes. We notice that ORE continues to train on the validation set, so that the model can leverage the information of unknown classes. In Tab. A8, we find ORE without training on valset (*i.e.*, froze parameters) obtains a rather lower $mAP_{\mathcal{K}}$ (53.96 *vs.* 58.45), and large amounts of known examples are misclassified to unknown. In contrast, OpenDet outperforms ORE without using the information of unknown classes.

Results on open world object detection. We also compare OpenDet with ORE in the task1 of open world object detection. As shown in Tab. A9, without accessing open-set data in the training set or validation set, OpenDet outperforms FR-CNN and ORE by a large margin and achieves comparable results with the Oracle.

E. Comparison with DS [11]

Implementation details. DS averages multiple runs of a dropout-enabled model to produce more confident predictions. As DS has no public implementation, we implement it based on the FR-CNN [12] framework. Specifically, we insert a dropout layer to the second-last layer of the classification branch in R-CNN, and set the dropout probability to 0.5. Previous works [1, 7] indicate that DS works even worse than the baseline method; we show it is effective as long as we remove the dropout layer during training, *i.e.*, we only use the dropout layer in the testing phase. Besides, original DS can only tell what is known, but do not have a metric for the unknown (*e.g.*, the unknown probability in OpenDet). We give DS the ability to identify unknown by entropy thresholding [5]. In detail, we define proposals with the entropy larger than a threshold (*i.e.*, 0.25) as unknown.

[1] https://github.com/JosephKJ/OWOD/issues?q=cannot+reproduce

Method	use unknown's annotation in train set	use unknown's annotation in val set	$WI_{\downarrow}$	$AOSE_{\downarrow}$	$mAP_{\mathcal{K}\uparrow}$
FR-CNN (Oracle)	✓	×	4.27	6862	60.43
FR-CNN	×	×	6.03	8468	58.81
ORE	×	✓	5.11	6833	58.93
OpenDet	×	×	**4.44**	**5781**	**59.01**

Table A9. **Results on open world object detection [7].**

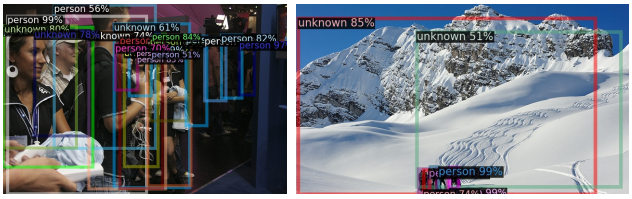
Method	#runs	$WI_{\downarrow}$	$AOSE_{\downarrow}$	$mAP_{\mathcal{K}\uparrow}$	$AP_{\mathcal{U}\uparrow}$
FR-CNN	1	18.39	15118	58.45	0
DS	1	15.26	18227	56.60	5.67
	3	16.41	14593	57.88	5.48
	5	16.76	13862	57.98	5.31
	10	16.91	13327	58.24	4.97
	30	16.98	12868	58.35	5.13
	50	17.01	12757	58.29	4.94
OpenDet	1	**14.95**	**11286**	**58.75**	**14.93**

Table A10. **Comparison with DS [11].** #runs denotes the number of runs used for ensemble. The row with gray background is reported in our main paper.

DS with different #runs. DS requires multiple runs for a given image. We report DS with different number of #runs in Tab. A10. By increasing #runs, DS shows substantial improvements on AOSE and $mAP_{\mathcal{K}}$, while the performance on WI becomes worse. We report DS with 30 #runs in our main paper, which is consistent with its original paper [11].

F. More Qualitative Results.

Fig. A3 gives more qualitative comparisons between the baseline method and OpenDet. OpenDet can recall unknown objects from known classes and the "background". Besides, we also give two failure cases in Fig. A2. **(a)** We find OpenDet performs poorly in some scenes with dense objects, *e.g.*, images with lots of `person`. **(b)** OpenDet classifies "real" background to the `unknown` class.



(a) (b)

Figure A2. **Failure cases.**

References

[1] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *WACV*, pages 1021–1030, 2020. 1, 3

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[5] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 3

[6] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *CVPR*, pages 1175–1184, 2021. 1

[7] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021. 1, 2, 3

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

[9] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. *arXiv preprint arXiv:2104.02939*, 2021. 1

[10] 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. *arXiv preprint arXiv:2103.14030*, 2021. 2, 3

[11] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *ICRA*, pages 3243–3249. IEEE, 2018. 2, 3

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, pages 1137–1149, 2017. 2, 3

[13] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE TPAMI*, 35(7):1757–1772, 2012. 1

[14] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *CVPR*, pages 13480–13489, 2020. 1

[15] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021. 1

[16] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021. 1, 2

Figure A3. **More qualitative comparisons** between the baseline and OpenDet.