

# OpenRSD: Towards Open-prompts for Object Detection in Remote Sensing Images

## Supplementary Material

This supplementary material provides more details about training strategy in Sec. 1, experiments about open-set object detection in Sec. 2, and more qualitative analysis in Sec. 3.

### 1. Training Strategy

Tab. 1 presents the empirical sampling rates for each sub-dataset within ORSD-Pre and ORSD+ dataset during training. These sampling rates are normalized into a probability distribution for random sampling at each training iteration. This approach helps mitigate severe sample distribution imbalances and variations in task complexity across RS datasets.

Tab. 2 compares the impact of different sampling rates on training performance, including balanced and the empirical sampling rates. Balanced sampling refers to randomly sampling a dataset for training in each iteration according to a uniform distribution. While the overall average performance remains nearly the same across different sampling strategies, significant variations are observed in individual dataset performance. Compared to empirical sampling, balanced sampling improves performance by 1.1% on DIOR-R

Table 1. The datasets used in the pretraining and fine-tuning stages, along with their respective sampling rates.

Name	Datasets	Sampling Rates
ORSD-Pre	DOTA-v2.0, DIOR-R, FAIR1M-2.0, SpaceNet, Xview, HRSC2016, GLH-Bridge, Million-AID	8: 2: 8: 2: 2: 1: 4: 16
	DOTA-v2.0, DIOR-R, FAIR1M-2.0, SpaceNet, Xview, HRSC2016, GLH-Bridge, fMoW, WHU-Mix, ShipRSImageNet	8: 2: 8: 2: 2: 0.5: 1: 4: 2: 0.5

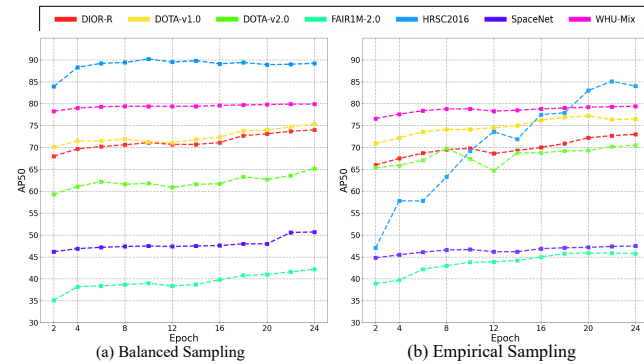


Figure 1. The convergence curves of empirical and balanced sampling rates across various datasets.

Table 2. Ablation study of dataset sampling rates.

Sampling Rate	DIOR-R	DOTA-v1.0	DOTA-v2.0	FAIR1M-2.0
Balance	74.0	75.3	65.2	42.2
Emperical	72.9	76.6	70.4	45.9
	WHU-Mix	SpaceNet	HRSC2016	Average
Balance	79.9	50.7	89.2	68.1
Emperical	79.4	47.5	84.6	68.2

Table 3. Open-set object detection performance on the STAR [2] dataset, including recall and precision for the categories 'Car,' 'Tool Gate,' and 'Cooling Tower,' as well as overall average recall (AR50) and AP50. Models marked with \* are trained using the proposed ORSD+ dataset.

Methods	Car		Tool Gate		Cooling Tower		AR50	AP50
	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.		
CastDet*	89.7	49.5	0.0	0.0	0.0	0.0	23.2	10.7
PKINet*	82.7	48.1	0.0	0.0	2.9	0.0	25.9	10.8
OpenRSD (Text)	<b>98.2</b>	<b>54.2</b>	47.4	<b>6.8</b>	80.0	0.1	40.0	13.4
OpenRSD (Image)	98.0	<b>54.2</b>	<b>68.4</b>	0.1	<b>85.7</b>	<b>66.5</b>	<b>45.3</b>	<b>13.5</b>

and 4.6% on HRSC2016. However, it results in decreases of 3.7% and 5.2% on DOTA-v2.0 and FAIR1M-2.0, primarily due to the varying number of iterations required for convergence across different datasets. Fig. 1 illustrates the convergence curves of different datasets under balanced and empirical sampling strategies. As illustrated in Fig. 1(a), we observe significant differences in the number of iterations required for convergence across datasets. For the large-scale FAIR1M-2.0 dataset, balanced sampling fails to achieve full convergence, whereas smaller datasets, such as HRSC2016 and WUH-Mix, rapidly reach their performance upper bounds. Empirical sampling allows the model to learn more effectively from large-scale datasets, making it better suited for real-world applications. Investigating dynamic sampling strategies will be our future work.

Additionally, we use CLIP to filter the pseudo labels. Using appropriate CLIP score and confidence score threshold significantly improves pseudo-label quality. Without this strategy, self-training performance falls from 69.3% to 68.7%.

### 2. Open-set Object Detection

We evaluate the open-set object detection performance. Since our ORSD+ dataset already encompasses most common RS object categories, using existing small-scale RS OVD settings for evaluation would be unfair. Therefore, we employ the newly released STAR [2] dataset for evaluation. The STAR dataset is a large-scale scene graph generation

dataset covering 11 complex geospatial scenarios closely related to human activities, including airports, ports, nuclear power plants, and dams. It contains 58 categories, nearly half of which are not included in the ORSD+ dataset. We compare two detection methods, CastDet\* and PKINet\*, both trained on the ORSD+ dataset.

Tab. 3 reports the recalls and precisions for the categories ‘Car,’ ‘Tool Gate,’ and ‘Cooling Tower,’ as well as the overall average recall (AR50) and AP50. For the ‘Car’ category, which appears in ORSD+, all methods achieve effective detection, with OpenRSD demonstrating superior performance. However, for ‘Tool Gate’ and ‘Cooling Tower,’ which are novel categories unseen during training, CastDet\* and PKINet\* struggle to detect them. For the ‘Tool Gate’ category, OpenRSD achieves a high recall rate but fails to distinguish it effectively from existing categories, leading to low precision. For the ‘Cooling Tower’ category, OpenRSD successfully detects objects based on image prompts, whereas text prompts fail. This discrepancy may stem from the stronger generalization of visual feature associations, while the sparse semantic relationships in RS categories limit the effectiveness of text prompts. Overall, visual prompts yield higher recall and are more beneficial for detecting unseen categories.

### 3. Prompt strategy

Prompts that are semantically aligned with the image tend to yield better results. Compared to existing prompt-based detection methods, OpenRSD supports a far more diverse set of prompt modalities and content - including mixed, multiple, and large-vocabulary prompts - greatly enhancing its practical applicability.

Additional visualization results are presented in Fig. 2, Fig. 3, and Fig. 4. Fig. 2 and Fig. 3 compare the detection performance before and after self-training under different prompt settings. After self-training, the model better handling mixed prompts across various scenarios, enhancing both recall and precision. The improvement is particularly pronounced in dense small-object detection, as self-training incorporates extensive annotations of relevant scenes.

Fig. 4 compares the effects of image and text prompts in different scenarios. Compared to text prompts, image prompts provide richer visual information, making it easier to distinguish fine-grained categories. As shown in the first column of Fig. 4, the image prompt successfully detects the ‘Tower-Crane’. In the second column, it helps avoid misclassifying objects as belonging to the ‘Dam’ category.

However, image prompts can also bring misclassifications. As illustrated in the third and fourth columns of Fig. 4, image prompts mistakenly classify some ships as vehicles due to their visual similarity. In contrast, text prompts offer a more explicit semantic distinction between ships and vehicles, reducing category confusion.

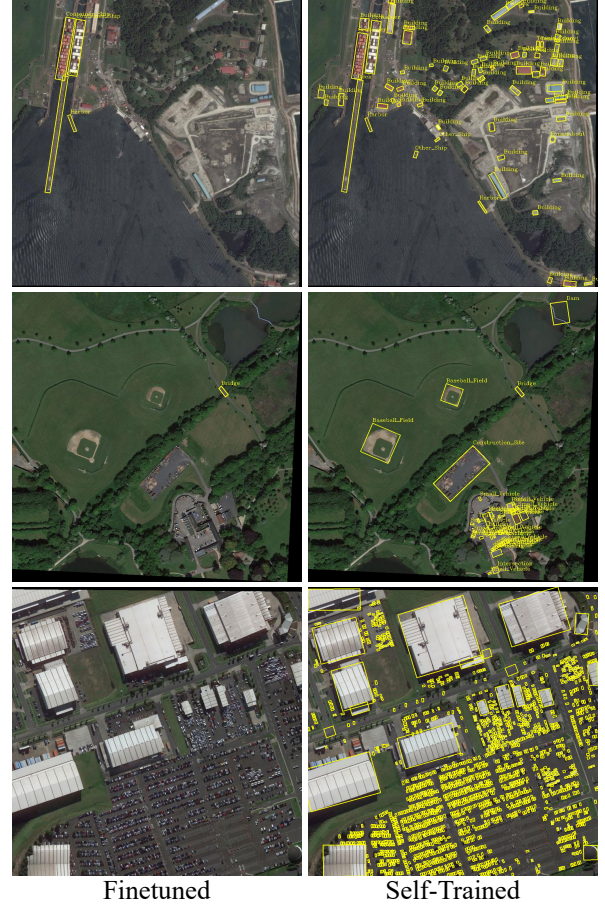


Figure 2. Visualization results under the ‘detect anything’ prompt on the DOTA-v2.0 [1] validation set before (the left column) and after self-training (the right column).

We conducted an ablation in which we replaced our diverse prompts with simple class name prompt. This change caused AP50 to fall slightly from 69.3% to 69.0%. Conversely, when the single-prompt-trained model was evaluated using our full suite of diverse prompts, its AP50 collapse to 59.6%.

### References

- [1] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2, 3
- [2] Yansheng Li, Linlin Wang, Tingzhu Wang, Xue Yang, Junwei Luo, Qi Wang, Youming Deng, Wenbin Wang, Xian Sun, Haifeng Li, et al. Star: A first-ever dataset and a large-scale benchmark for scene graph generation in large-size satellite





Figure 3. Visualization results on the DOTA-v2.0 [1] validation set before (the top row) and after self-training (the bottom row), demonstrating three prompts: detecting ship, detecting vehicle, and detecting buildings.

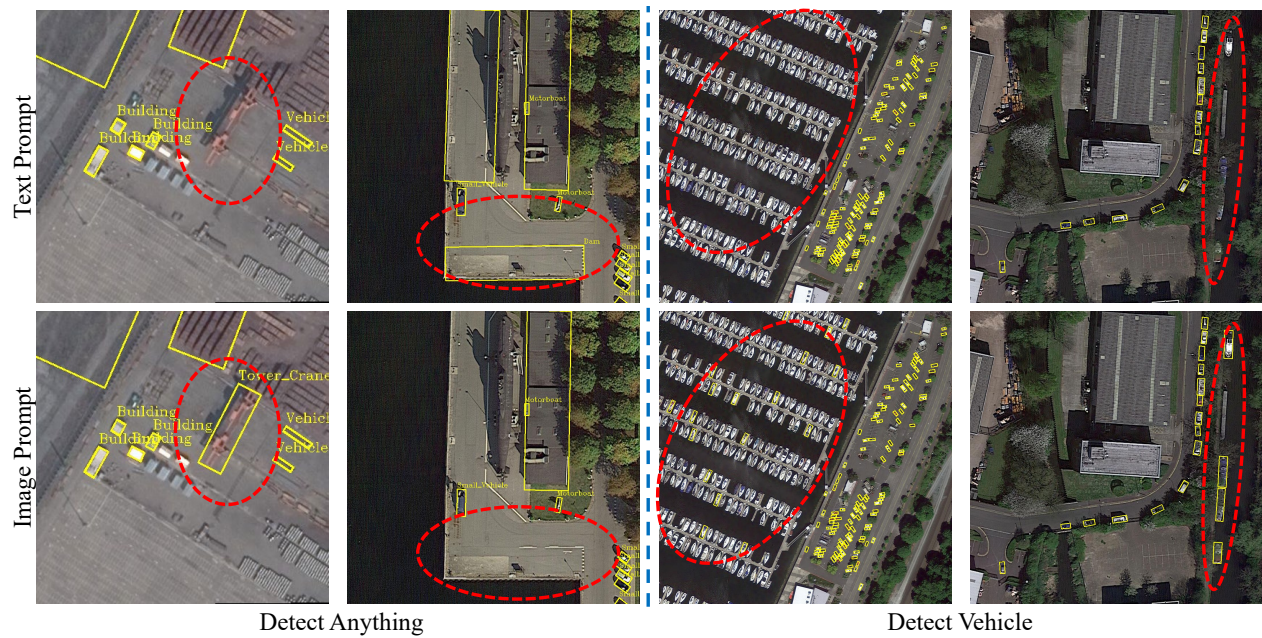


Figure 4. Visualization results on the DOTA-v2.0 [1] validation set using image and text prompts, demonstrating two prompts: detecting any objects and detecting vehicle. The red circles highlight the differences.