Open-Set Object Detection By Aligning Known Class Representations
Supplementary Material

Hiran Sarkar1 Vishal Chudasama1 Naoyuki Onoe1 Pankaj Wasnik1* Vineeth N Balasubramanian2
1Sony Research India 2Indian Institute of Technology Hyderabad
{hiran.sarkar,vishal.chudasama1,naoyuki.onoe,pankaj.wasnik}@sony.com, vineethnb@cse.iith.ac.in

This supplementary presents the following details which we could not include in the main paper due to space constraints:

Contents
S1 Experimental Settings
   S1.1 Dataset Details . . . . . . . . . . . . . . . . . . . . . . .
   S1.2 Implementation Details . . . . . . . . . . . . . . .
   S1.3 Generation of centerness targets . . . . . . . .
S2 Ablation Studies & Analysis
   S2.1 Effect of prompts in Semantic Clustering module . . . . .
   S2.2 Effect of different thresholds in entropy thresholding evaluation mechanism . . . . .
S3 Experimental Results
   S3.1 Additional Results . . . . . . . . . . . . . . . . . . . . .
   S3.2 Comparison on OWOD setting . . . . . . . . . .
   S3.3 Additional Qualitative Results . . . . . . . . . .
   S3.4 Failure Case Analysis . . . . . . . . . . . . . . .

S1. Experimental Settings
S1.1. Dataset Details

We used the Pascal VOC [1] and MS-COCO [5] for training and testing purposes. This section presents more details about these datasets and the open-set object detection (OSOD) based evaluation settings.

Pascal VOC [1]: It contains VOC07 trainval set having 5,011 images, and VOC12 trainval set having 11,540 images with 20 labeled classes. Further, VOC07 val split set is taken as a validation dataset.

MS-COCO [5]: This dataset comprises a training set of more than 118,000 images with 80 labeled classes. While the validation dataset (val2017) contains 5000 labeled images.

**Corresponding author**

S1.2. Implementation Details

In addition to experimental analysis on ResNet50 and ConvNet backbone presented in main manuscript, we present further analysis on Swin-T backbone [6]. To do such experiments, we have opted to utilize AdamW as an optimizer with a learning rate of 1e-4 and trained for 32,000 iterations with a 0.05 weight decay rate during training phase. The training process has been facilitated by a single GPU with a batch size of 6. For fair comparison, we re-train the Faster R-CNN [9], DS [7], PROSER [11] and OpenDet [3] methods on same configuration.

Open World Object Detection (OWOD) setting: To demonstrate how the proposed method performs in the

The process of closed-set training is executed on VOC07 trainval and VOC12 trainval set. While the close-set performance is evaluated on the test split of VOC07. For testing under open-set conditions, we follow the evaluation protocol suggested in [3] where testing images having 20 VOC classes and 60 non-VOC classes [5] are employed and categorized in two settings named as VOC-COCO-T1 and VOC-COCO-T2.

- **VOC-COCO-T1**: In this setting, the 80 COCO classes have been categorized into four groups, each comprising 20 classes, based on their semantics [3, 4]. To create VOC-COCO-\{20, 40, 60\}, we utilized 5000 VOC testing images and \{n, 2n, 3n\} COCO images, each of which contained \{20, 40, 60\} non-VOC classes with semantic shifts, respectively.

- **VOC-COCO-T2**: In this setting, four datasets have been constructed by gradually increasing the wilderness ratio while utilizing n = 5000 VOC testing images and \{0.5n, n, 2n, 4n\} COCO images, disjointing with VOC classes. Unlike the VOC-COCO-T1 setting, the VOC-COCO-T1 aims to assess the model’s performance under significantly greater wilderness, whereby a substantial quantity of testing instances remain unseen during the training process.
context of OWOD setting, we conducted evaluation according to the ORE protocol \cite{4}, which was specifically designed for OWOD and comprises four tasks aimed at assessing the performance of OSOD and incremental learning. However, as our work is not concerned with incremental learning, we restrict our evaluation to task 1. The dataset utilized for task 1 comprises 16551 Pascal VOC images with 20 classes \cite{1} for training and the 10246 testing images having 20 VOC classes and 60 COCO classes \cite{5} for open-set evaluation. Here, we compare the proposed method against the baseline Faster R-CNN \cite{9} and its oracle version\footnote{https://github.com/JosephKJ/OWOD}, in addition to OWOD methods (ORE \cite{4}, OW-DETR \cite{2}, PROB \cite{13}) and OSOD methods (OpenDet \cite{3} and Openset RCNN \cite{12}).

S1.3. Generation of centerness targets

This section presents the procedure of generating the centerness target, i.e., $C\text{targets}$, for calculating the centerness loss. The corresponding steps are mentioned below.

- The initial step involves the conversion of the default ground-truth bounding box and proposal coordinates, which are in $(x_1, y_1, x_2, y_2)$ format, to $\{cx, cy, h, w\}$ format. This conversion results in the center coordinates represented by $cx$ and $cy$, while $h$ and $w$ represent the height and width of the bounding box or proposal, respectively. In the case of a ground-truth box $i$ and proposal box $j$, the transformed bounding box and proposal targets can be denoted by $\{cx_{gt(i)}, cy_{gt(i)}, h_{gt(i)}, w_{gt(i)}\}$ and $\{cx_{p(j)}, cy_{p(j)}, h_{p(j)}, w_{p(j)}\}$, respectively.

- Subsequently, the differences in those quantities between the proposal box $j$ and the ground truth boxes are determined. Concerning the ground-truth box $i$, the differences can be computed as follows.
  
  \[
  dx_{ij} = \frac{cx_{gt(i)} - cx_{p(j)}}{w_{p(j)}} \\
  dy_{ij} = \frac{cy_{gt(i)} - cy_{p(j)}}{h_{p(j)}} \\
  dw_{ij} = \log\left(\frac{w_{gt(i)}}{w_{p(j)}}\right) \\
  dh_{ij} = \log\left(\frac{h_{gt(i)}}{h_{p(j)}}\right)
  \]

  Here, we filter out the targets with negative values. Finally, the centerness target for proposal box $j$ and ground-truth box $i$ can be calculated as given in \cite{10}.

  \[
  C_{target} = \sqrt{\min(dx_{ij}, dy_{ij}) \cdot \min(dw_{ij}, dh_{ij})} / \max(dx_{ij}, dy_{ij}) \cdot \max(dw_{ij}, dh_{ij})
  \]

  where, $\min(\cdot)$ and $\max(\cdot)$ denote the minimum and maximum operations.

S2. Ablation Studies & Analysis

This section presents additional ablation analysis to establish the efficacy of the proposed framework. All ablation experiments are trained using ConvNet backbone to ensure a fair comparison and evaluated on the VOC-COCO-40 setting.

S2.1. Effect of prompts in Semantic Clustering module

In the proposed framework, we have introduced a semantic clustering module that utilizes a CLIP-based text encoder \cite{8} to generate a 1024-dimensional text embedding. In contrast to the original CLIP approach \cite{8} that uses a single prompt, seven prompts, or 80 prompts, we have utilized the class name as the prompt. To see the impact of using only the class name as a prompt, we conducted several ablation experiments in which the proposed framework is trained with different prompts in the semantic clustering module. The corresponding findings, depicted in Figure S1, indicate that the proposed framework with only class name as prompt performs better than other settings in terms of $mAP_k$, $AP_u$, and HMP metrics.

![Figure S1. Effect of different prompts in CLIP-based text encoder of semantic clustering module on VOC-COCO-40 setting.](image)

S2.2. Effect of different thresholds in entropy thresholding evaluation mechanism

Figure S2 depicts the impact of varying threshold values for entropy thresholding. Reducing the threshold value results in a decrease in the number of misclassified unknown instances, which leads to an improvement in AOSE. Simultaneously, the $WI$ is also improved by decreasing the

\footnotetext[1]{https://github.com/JosephKJ/OWOD}

\footnotetext[2]{An ‘Oracle’ detector is a reference model that has access to all known and unknown labels at any given point \cite{4}.}
Table S1. Comparison with SOTA methods on VOC-COCO-T1 setting on Swin-T backbone. The best-performing measures are highlighted with bold font while the second-best is highlighted with underlined italic font. * indicates the re-trained methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC-COCO-n</th>
<th>VOC-COCO-2n</th>
<th>VOC-COCO-4n</th>
<th>VOC-COCO-6n</th>
<th>VOC-COCO-8n</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>AOSE↑</td>
<td>AP↑</td>
<td>HMP↑</td>
<td>W</td>
</tr>
<tr>
<td>Faster RCNN* [9]</td>
<td>78.74</td>
<td>11.39</td>
<td>21562</td>
<td>57.21</td>
<td>0.00</td>
</tr>
<tr>
<td>DST* [1]</td>
<td>78.08</td>
<td>9.58</td>
<td>16769</td>
<td>57.81</td>
<td>7.51</td>
</tr>
<tr>
<td>Our (proposed)</td>
<td>79.27</td>
<td>8.12</td>
<td>10667</td>
<td>58.87</td>
<td>16.93</td>
</tr>
</tbody>
</table>

Table S2. Comparison with SOTA methods on VOC-COCO-T2 setting on Swin-T backbone. The best-performing measures are highlighted with bold font while the second-best is highlighted with underlined italic font. * indicates the re-trained methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>VOC-COCO-n</th>
<th>VOC-COCO-2n</th>
<th>VOC-COCO-4n</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>AOSE↑</td>
<td>AP↑</td>
</tr>
<tr>
<td>Faster RCNN* [9]</td>
<td>13.01</td>
<td>10941</td>
<td>70.31</td>
</tr>
<tr>
<td>OpenDet* [3]</td>
<td>9.21</td>
<td>8856</td>
<td>75.28</td>
</tr>
<tr>
<td>Our (proposed)</td>
<td>9.29</td>
<td>7383</td>
<td>74.04</td>
</tr>
</tbody>
</table>

Figure S2. Effect of different thresholds in entropy thresholding mechanism on VOC-COCO-40 setting.

threshold value. However, this reduction also coincides with a decrease in precision scores. The decline in $AP_u$ arises from the incompleteness of annotations in the COCO dataset, which results in numerous unknown predictions being classified as False Positives. As a result, there is a trade-off between achieving a favorable $AOSE$ score and maintaining a high precision score through entropy thresholding. We opt for a threshold of 0.85 for our experiments as it yields balanced performance across all metrics.

S3. Experimental Results

In addition to the experimental analysis presented in the main manuscript, we elaborate on some additional experimental results.

S3.1. Additional Results

In addition to result comparison with existing OSOD works on ResNet50 and ConvNet backbone, we have also compared results on Swin-T [6] backbone. In Table S1, we present a comparison on the VOC-COCO-T1 setting. Here, we can see that the proposed method improves $WI$ by 2 – 5% and $AOSE$ by 2000 – 3000 than previous best-performing PROSER [11] and OpenDet [3] results in all dataset settings. We also show improvements as high as 5% on $AP_u$ and 4% on $HMP$, on VOC-COCO-40 compared to previous best-performing results. In VOC-COCO-60, the proposed method obtains a better $AP_u$ score of 4.55 similar to PROSER [11]; however, it outperforms PROSER model in $mAP_u$ by a gain of 4.4%. Moreover, the comparison of VOC-COCO-T2 setting is presented in Table S2, where the proposed method performs better than other methods in terms of $AOSE$, $AP_u$ and $HMP$ metrics in all settings. We show a gain of 5–8% in $AP_u$,$A-6\%$ in $HMP$ and improves $AOSE$ by 1500 – 5500 than OpenDet [3].
Figure S3. Visual comparison between our proposed and other methods. (Zoomed-in for better view)
Due to limited space in our main paper, we also report the results on VOC-COCO-0.5n in Table S3 based on ResNet50, ConvNet and Swin-T backbones. Here, one can observe that the proposed method outperforms existing methods by a significant margin in all cases except in $mAP_k$ from the ResNet50 backbone-based comparison.

### S3.2. Comparison on OWOD setting

We evaluate the proposed method in the context of OWOD setting, i.e., task 1 as suggested in [4] and compare the results against existing methods, presented in Table S4. This analysis reveals that the proposed method performs better when employed with a ResNet50 backbone than other methods in terms of $WI$ and $AOSE$. Furthermore, when used with a ConvNet backbone, our proposed method improves the performance further and obtains significant performance than other methods.

Table S4. Comparison with OWOD based task 1 evaluation setting [4]. The best-performing measures are highlighted with **bold font.** † indicates results obtained from OpenDet [3] paper, while †† indicates results from Openset-RCNN [12] paper.

<table>
<thead>
<tr>
<th>Method</th>
<th>OWOD-Task-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN (Oracle)† [9]</td>
<td>$WI$ ↓, $AOSE$ ↓, $mAP_k$ ↑</td>
</tr>
<tr>
<td>Faster R-CNN† [9]</td>
<td>4.27 6862 60.43</td>
</tr>
<tr>
<td>ORE† [4]</td>
<td>5.11 6833 56.34</td>
</tr>
<tr>
<td>OW-DETR†† [2]</td>
<td>5.71 10240 59.21</td>
</tr>
<tr>
<td>PROB [13]</td>
<td>— — —</td>
</tr>
<tr>
<td>OpenDet [3]</td>
<td>4.44 5781 59.01</td>
</tr>
<tr>
<td>Openset-RCNN [12]</td>
<td>4.67 5403 59.34</td>
</tr>
<tr>
<td><strong>Our (ResNet50)</strong></td>
<td>3.76 5145 57.44</td>
</tr>
<tr>
<td><strong>Our (ConvNet)</strong></td>
<td>3.52 4616 61.51</td>
</tr>
</tbody>
</table>

### S3.3. Additional Qualitative Results

In addition to quantitative analysis, we have provided qualitative results in Figure S3 to demonstrate the improvement of our method over baseline methods such as Faster RCNN [9], PROSER [11] and previous best-performing OpenDet [3]. It can be visualized that the proposed method accurately classifies unknown objects that are semantically closer to known classes, which other methods fail to do. For example, Faster R-CNN [9] and PROSER [11] misclassify ‘goat’ as ‘cow’ due to their semantic similarity (see 1st row of Figure S3). However, our model, having learned semantic-based clusters, correctly labels ‘goat’ as the ‘unknown’ class. It can also be observed that other methods misclassify the ‘giraffe’ as either ‘cow’, ‘sheep’, ‘dog’ or ‘horse’ as depicted in 3rd row in Figure S3. In contrast, our proposed method accurately identifies it as ‘unknown’. Similarly, other models misclassify ‘bed’ as ‘sofa’ due to their semantic similarity. At the same time, the proposed method predicts it accurately as an unknown class (as illustrated in the last row in Figure S3).

### S3.4. Failure Case Analysis

In Figure S4, we present several instances where our model fails to perform well. The proposed framework detects false positive ‘unknown’ objects in all three images. We posit that this problem may arise due to a limitation of the object focus loss and its tendency to promote additional unknown detection. As a result, in certain cases, this mechanism may detect objects that are not even present.

Figure S4. failure cases.

### References


[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,


