Supplementary Material for Find n' Propagate: Open-Vocabulary 3D Object Detection in Urban Environments

Djamahl Etchegaray[1], Zi Huang[1], Tatsuya Harada[2], and Yadan Luo *[1]

[1] UQMM Lab, University of Queensland, Brisbane, Australia
[2] The University of Tokyo, Tokyo, Japan.
{uqdetche, helen.huang, y.luo}@uq.edu.au, harada@mi.t.u-tokyo.ac.jp

Overview

This supplementary material provides complementary details to better understand the main paper. A brief overview of the sections is available below:

- 1: Visualisation of Detection Results under §Setting 1 and §Setting 3
- 2: Further Implementation Details
  - 2.1: Filtering Strategies in Iterative Self-Training
- 3: More Ablation Studies
  - 3.2: Greedy Box Oracle
  - 3.3: Self-Training Objectives
  - 3.4: Top-down Projection
  - 3.5: Discussions on CLIP2Scene Zero-shot Results

1 Visualisation

To achieve a better understanding of the proposed approach, we provide more visualisations of qualitative analysis on the derived Box Seeker (column 2), Find n' Propagate (column 3) and Top-down self-training on §Setting 1 in Fig. 1. The top row gives the corresponding multi-view images of the studied point cloud. The base classes are highlighted in blue while the novel ones are colored in red. Points are coloured based on the class of the ground truth box they are in to help highlight which objects have been missed, if they do not fall in any ground-truth box their colour is grey. The figure shows the significant increase in the novel recall by using Find n' Propagate over a strong baseline Top-down self-train. Further, it establishes the number of new objects that were able to be found/refined in self-training. The detailed comparisons can be found in the captions of each figure.

§Setting 3. We previously evaluated our novel proposal generation on nuScenes, demonstrating the effectiveness on both base and novel classes. Visualisations of these proposals for all classes are available in Fig. 1 (column 2). Further to our

* Correspondence to Yadan Luo (y.luo@uq.edu.au)

self-training experiments with §Setting 1 and §Setting 2, we experimented with training on Setting 3 with the Box Seeker. Regrettably, our findings indicate that the proposed self-training technique struggled to effectively learn from the noisy proposals, primarily attributing this challenge to the inherent difficulty in assimilating semantic information from the noisy center heatmap. While some common objects were successfully identified, the majority remained elusive, resulting in a notably low precision. We intend to tackle this issue in our future research endeavors.

2 Implementation Details

In our study, we implement both Top-up and Bottom-down OV-3D approaches in the OpenPCDet[3] codebase. The source code is available in the supplementary material for reference. We use Transfusion [1] and Centerpoint [6] as the 3D detection backbones with the default hyperparameters. For the implementation of logit fusion in the Top-down approaches, the threshold γ is set to 0.2. For the search space in Greedy Box Seeker, we configure the number of intervals for orientation k_o at 10, scale k_s at 4, and depth k_d also at 4. In augmentation, the memory bank size $|\mathcal{Q}|$ for each class is 60. The training batch sizes are consistently fixed to 8. The Adam optimizer is adopted with a learning rate initiated as 0.001, and scheduled by the OneCycle scheduler. Different from [1, 6], we do not disable GT sampling for the last 5 epochs of training, as we found it would deteriorate the novel performance.

Top-down Clustering. For HDBScan we use a minimal cluster size of 15, and for DBScan an epsilon of 1.5. We utilise the sklearn package [4] in Python for extracting clusters followed by box estimation from [3].

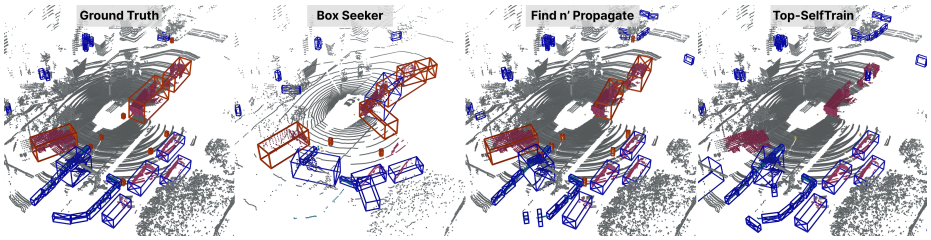
2.1 Filtering Strategies in Iterative Self-training

In this section, we elaborate on the filtering strategies in the last self-training step of the proposed Find n' Propagate framework. During iterative self-training it is necessary to filter and combine objects from multiple sources. Sources include Box Seeker proposals, self-training pseudo boxes, and Remote Propagator boxes to be pasted. Firstly, for all the sources we must remove any that overlap with GT, as detailed below in Filtering Novels. Secondly, once we filter out any novel boxes that overlap with GT, we must make sure that none overlap with each other. We utilise standard non-max suppression (NMS) with the pseudo scores to remove any boxes that overlap and keep only the ones that were well captured in self-training. Thirdly, we must remove novel proposals that are too close or too sparse, as these are likely to be low quality or false positives.

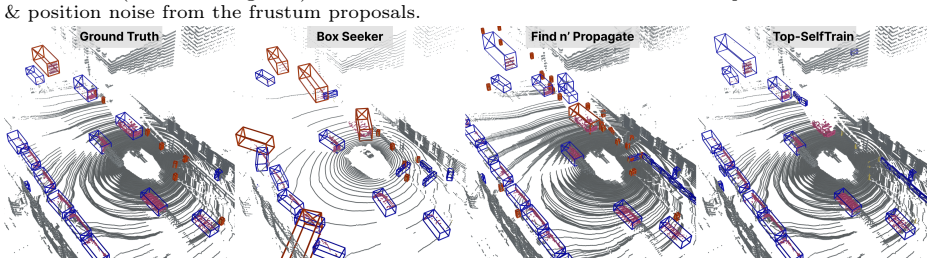
Filtering Novels. Specifically, we remove erroneous novel instances, $\{\mathcal{Q}_i\}_{i=1}^{|\mathcal{Q}|}$, that overlap with GT base annotations, $\{\mathfrak{B}_j^{\mathrm{B}}\}_{j=1}^{M}$, by calculating the maximum IoU with any base GT, and removing those above a threshold, β_{overlap}:

$$\texttt{remove}(\mathcal{Q}_i) = \left(\texttt{max}_{k\in|\mathcal{Q}|}\texttt{IoU}(\mathfrak{B}_j^{\mathrm{Base}}, \mathcal{Q}_i)\right) > \beta_{\mathrm{overlap}}.$$

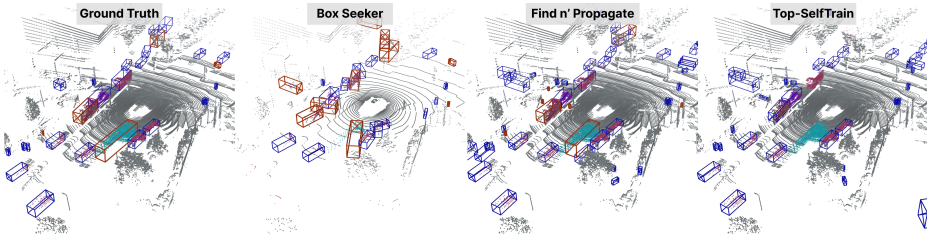
[3] https://github.com/open-mmlab/OpenPCDet



(a) None of the novel instances can be found in this scene with Top-Down self-train, but Box Seeker and Find n' Propagate learning can sufficiently localise them. The frustum proposals can localise many novel and base objects, however, the localisation is noisy as can be seen with the varied orientations (camera facing bias). The Find n' Propagate model is able to improve the orientation & position noise from the frustum proposals.



(b) In this scene the recall is also significantly better for Find n' Propagate, however, the detector has produced a lot of traffic cones to maximise their recall. We can see that a number of them are true positives. Unfortunately, the frustum proposals contain a lot of misclassification errors from GLIP, where many base classes are incorrectly mapped to novel classes.



(c) Top-down self-train can suffer from poor base performance, here Top-selftrain with CLIP is unable to find the large vehicle towards the centre of the point cloud. Like previous visualisations, no novels are proposed for this scene by Top-down Self-train.

Fig. 1: Visualisation of the Ground Truth, Greedy Box Seeker, Find n' Propagate and Top-down Self-Train. Predicted boxes are coloured blue for base classes and red for novel classes, under § Setting 1. Points are coloured based on the class of the ground truth box they are in to help highlight which objects have been missed, if they do not fall in any ground-truth box their colour is grey. The base ground truth and novel Box Seeker proposals are utilised for Find n' Propagate self-training. The base Seeker proposals are shown for visualisation only, as in reality Box Seeker is under § Setting 3 and has no base classes it has learnt from. The effectiveness of our Find n' Propagate instance learning is highlighted, as Top-selftrain misses many novel and some base objects but ours is able to improve the recall drastically without poor regression quality.

We empirically set $\beta_{\text{overlap}} = 0.1$ to allow some overlap as the frustum proposals often have erroneous orientations.

Filtering Too Sparse and Too Close Objects. During self-training, we maintain a queue of $|\mathcal{Q}|$ novel samples, which can be derived from both Box Seeker proposals and pseudo-labels. To update the queue for each batch, we rank boxes by their confidence, then filter out any that have *too few* points or occur *too close* to the ego vehicle. As we have already filtered novel instances that overlap with known GT annotations, then the high-confidence predictions should be unknown objects that are newly discovered.

Combining Seeker Proposals and Self-Training Pseudos. Iterative self-training must integrate proposals from Box Seeker and high-confidence pseudo-boxes from self-training. To do this, we concatenate them together and perform NMS. The Box Seeker utilises the VLM score as the confidence score since the quality control score is local to each frustum. As the VLM score is usually quite high, the self-training pseudo has to be very confident to replace the frustum proposal.

3 More Ablation Studies

Due to space limitations in the main paper, we include further ablations for all the proposed modules. We evaluate the effectiveness of the Region VLM (Sc. 3.1), Greedy Box Oracle (Sec. 3.2): the values of the alignment criteria coefficient, search space depth quantiles and search space size sensitivity. Then, we study the design choices in the Sec. 3.3 self-training module, including loss normalisation/reweighting. Finally, we supply the details for our Top-down comparisons (Sec. 3.4), providing the steps for fusing the labels from 3D proposals and VLM predictions.

3.1 Region VLM

The proposed approach is agnostic to region VLM choices. We compared the quality of the Greedy Box Seeker with various 2D prediction sources in Table 1. The GT_{2D} serves the upper bound of the performance. Cameras capture close and dense objects, missing many faraway ones whilst the GT_{2D} captures every labelled object. However, 3D annotations are projected to generate GT_{2D} and often do not tightly bound the 2D view of the object. Due to this, both OWL-ViT and GLIP can achieve *better* results: both surpass GT for Cars and Buses classes and GLIP is better for Construction Vehicles, Trucks and Motorcycles. GLIP and OWL-ViT can achieve surprising performance, with GLIP being 28% off the GT_{2D} mAP, finding most nearby objects.

3.2 Greedy Box Oracle

The following ablations discuss the effect of different frustum proposal generation components. For all settings, AP_{B} is defined for 6 classes known where the evaluation is done directly on frustum proposals.

Table 1: Evaluation on nuScenes with 10 novel classes (§ SETTING 3).

Box$_{3D}$	VLM	Novel										Overall	
		Car	Const.	Trai.	Barr.	Bic.	Ped.	Truck	Bus	Motor.	Cone.	AP$_N$	NDS
Seeker	OWL	**25.14**	0.76	0.00	0.30	19.21	12.24	6.07	5.89	18.01	18.89	10.65	18.30
	GLIP	24.28	**4.14**	**0.15**	**4.39**	**34.65**	**22.80**	**8.58**	**11.09**	**35.82**	**21.26**	**16.72**	**22.40**
	GT$_{2D}$	16.09	3.26	0.37	34.28	38.22	39.03	7.41	5.54	30.33	56.45	23.10	22.83

Impact of Alignment Criterion Coefficient. The impact of the alignment coefficient is detailed in Fig. 2, where we evaluate the Box Seeker proposal generation and Box Oracle through different $\alpha_{\text{IoU}} \in \{0, 1, 2, 3, 4\}$ under § Setting 3. The alignment criteria improves proposal filtering over density alone, but increasing the coefficient to 2 has a significant benefit, establishing that the 2D alignment provides better guidance for finding novel instances. However, increasing the value further (*e.g.*, 3-4) results in diminished performance as the density can only contribute to ranking by an insignificant amount.

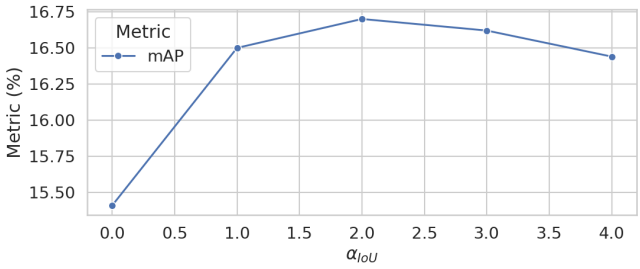


Fig. 2: Impact of Coefficient α_{IoU} under § Setting 3.

Impact of Search Space Quantiles. The quantile selection for d_{min} and d_{max} is crucial to separating between background and foreground in each frustum. The intuition is to select a large enough lower quantile to exclude any noise before the object, and a small enough upper quantile to remove any background noise. In Fig. 3 the lower quantiles $0.0, 0.25, 0.5$ and the upper quantiles $0.25, 0.5, 0.75$ are tested. Clearly, utilising a frustum with d_{min} and d_{max} from the quantiles 0.0 and 0.25 results in the best performance, as the background is sufficiently removed. Points closer to the camera than the object of interest do not significantly affect the proposal quality, as a lower quantile of 0.0 results in the best mAP. Overall, increasing the upper quantile results in worse performance as the background is not removed.

Search Space Sensitivity. We ablate the Greedy Box Seeker search space in Fig. 4. We experiment with varying the number of trials for each dimension, including k_d depths, k_o orientations and k_s sizes. We find that we can considerably improve computational efficiency whilst maintaining strong performance. The number of orientations, k_o can be reduced to 3, the number of depths can

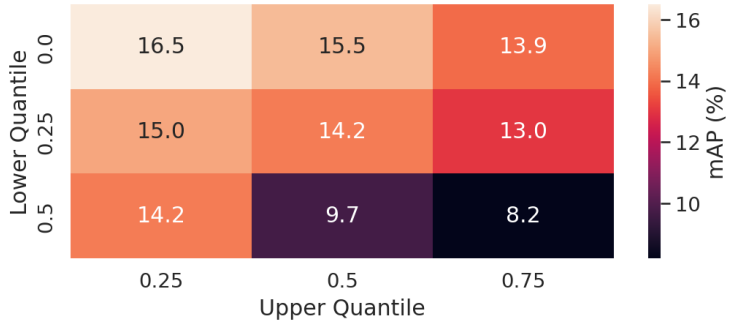


Fig. 3: Impact of Quantiles with respect to base and novel instance detection under § SETTING 3.

be reduced to 2 and the number of sizes 1, significantly reducing the number of trials from 320 (maximum) to 6.

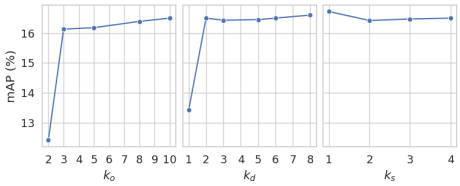


Fig. 4: Impact of k_o, k_d, and k_s.

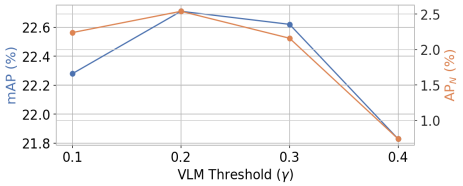


Fig. 5: Impact of Logit Fusion. Evaluated with Centerpoint backbone using GLIP and varying γ.

3.3 Self-Training Objectives

In the following section, we provide ablations for our self-training process. We measure the impact of introducing previous work on loss normalisation to mitigate the effect of label noise.

Impact of Self-Training Loss Normalisation. We tried to utilise prior work [7] on self-training to smooth the label noise from novel proposals on Centerpoint. The motivation for the loss is to balance the contributions of the supervised (base classes) and self-supervised (novel classes) during training, by using the exponential moving average (EMA) of the losses to re-weight the self-supervised part. The ratio between the EMA loss on base classes and novel classes is utilised to calculate a coefficient for the novel class loss, with a hyperparameter α to tune the weight of novel classes. Given the loss on base classes L_B and novel classes L_N, the loss is calculated as:

$$L = \frac{1}{1+\alpha}(L_B + \alpha \frac{\bar{L}_B}{\bar{L}_N} L_N),$$

where $\bar{L}_B$ and $\bar{L}_N$ are exponential moving averages. The loss was applied to the Centerpoint heatmap classification loss. Table 2 displays the results with and without loss normalisation. For both experiments, $\alpha = 0.5$ reduces the impact of noisy proposals by weighting the novel classes as half the base. There is mostly negligible difference in metrics, but loss normalisation gives a small absolute increase in AP_N of 0.69%.

Table 2: Open-vocabulary evaluation with Centerpoint and 4 novel classes (§ Setting 1) during training.

Loss Normalisation	mAP	NDS	AP_B	AP_N
	37.38	**40.28**	**49.99**	18.46
✓	37.38	39.82	49.54	**19.15**

3.4 Ablation Study on Top-down Projection

Impact of Logit Fusion. For top-down OV-3D baselines, we test different variants by incorporating the following strategies of logit fusion. During test-time, Top-down Projection methods must decide whether to relabel proposals as unknowns or keep them as the proposed known class. To do this, we set a threshold on the VLM confidence for each 3D proposal, and reject the VLM predicted label if the confidence is not sufficient. The outgoing label for proposal i, is given as $\widehat{\mathbf{Y}}_i$,

$$\widehat{\mathbf{Y}}_i = \begin{cases} \widehat{\mathbf{Y}}_i^{\text{3D}} & \mathbf{p}_i^{\text{VLM}} \leq \gamma \\ \widehat{\mathbf{Y}}_i^{\text{VLM}} & \text{otherwise} \end{cases}. \tag{1}$$

As shown in Fig. 5, a VLM confidence threshold of 0.2 provides the best performance balance, and also the highest AP_N.

3.5 Discussions on CLIP2Scene Zero-shot Results

Table 3: Zero-shot Lidar Segmentation on nuScenes with CLIP2Scene. Results are given in %. Row 1 shows the segmentation results yielded by CLIP2Scene, then rows 2 and 3 show the proposal results with the proposed Top-Clustering.

Task (Metric)	Car	Const.	Trai.	Barr.	Bic.	Ped.	Truck	Bus	Motor.	Cone.
Zero-shot Segmentation (mIoU)	29.60	9.90	0.00	7.90	1.30	0.60	45.60	43.90	9.40	18.70
Top-Clustering & HDBScan (AP_{3D})	4.01	0.00	0.00	0.00	0.00	0.00	1.38	0.37	0.00	1.27
Top-Clustering & DBScan (AP_{3D})	3.07	0.00	0.00	0.00	0.00	0.00	1.86	0.72	0.00	0.00

In Table 3, we show the zero-shot segmentation results (row 1) and the detection results with Top-Clustering as presented in the main paper. The

zero-shot segmentation results (mIoU) are below 10% for many classes, including `Construction Vehicle`, `Trailer`, `Barrier`, `Pedestrian` and `Motorcycle`; for these classes the detection result is **zero**, likely due to having an **inferior** pixel-wise classification accuracy, restricting the density of points with the correct label and thus making density clustering unable to find the correct objects. Visualisation of the misclassification error is available in Fig. 6, where it is clear that the road is associated with pedestrians and therefore segmented incorrectly. Objects with better segmentation accuracy have a likely chance of being correctly partitioned into clusters corresponding to individual object instances, as is the case for the rest of the classes. However, the average precision of these classes is quite low, due to the sparsity of points in a single sweep.

The lack of AP with this method suggests that the LiDAR scenes in nuScenes are probably too sparse for density-based clustering to work. We utilised 1 sweep for computational efficiency, and due to CLIP2Scene being trained on 1 sweep, further study is possible with LiDAR densification methods (aligning the sweeps) so that it is possible to run these methods with denser point clouds. We compared with 1 sweep to make it fair to our work, which only uses 1 sweep. Multi-sweep necessitates scene flow prediction and tracking, as done in previous works [2, 5] however these works consider only vehicles at this stage and need to be expanded to accomodate open vocabularies.

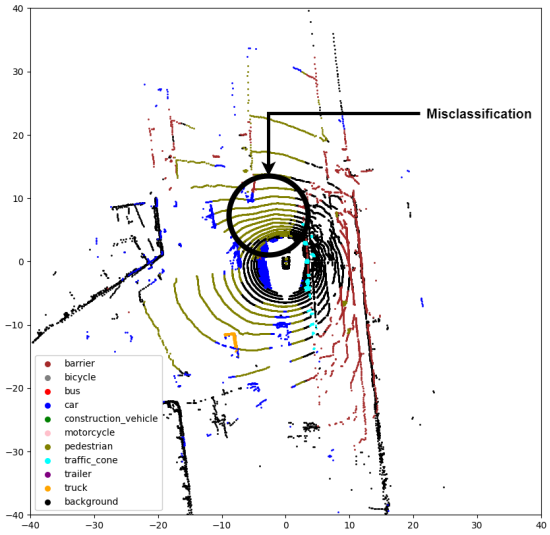


Fig. 6: CLIP2Scene Zero-shot misclassification. Many points on the road surface are incorrectly predicted as pedestrian.

References

1. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.: Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1080–1089. IEEE (2022)
2. Li, X., Pontes, J.K., Lucey, S.: Neural scene flow prior. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 7838–7851 (2021)
3. Lu, Y., Xu, C., Wei, X., Xie, X., Tomizuka, M., Keutzer, K., Zhang, S.: Open-vocabulary point-cloud object detection without 3d annotation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1190–1199 (2023)
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
5. Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D.: Offboard 3d object detection from point cloud sequences. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6134–6144. Computer Vision Foundation / IEEE (2021)
6. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11784–11793 (2021)
7. Zoph, B., Ghiasi, G., Lin, T., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems (NeurIPS) (2020)