Approaching Outside: Scaling Unsupervised 3D Object Detection from 2D Scene
Supplementary Material

Ruiyang Zhang[1], Hu Zhang[2], Hang Yu[3], and Zhedong Zheng[1]*

[1] FST and ICI, University of Macau, China
[2] CSIRO Data61, Australia
[3] Shanghai University, China
ruiyang.061x@gmail.com, Hu1.Zhang@csiro.au, yuhang@shu.edu.cn, zhedongzheng@um.edu.mo
https://github.com/Ruiyang-061X/LiSe

A Appendix

A.1 Implementation Details

In this section, we present more implementation details of LiSe.

Pseudo-label Generation (see Section 3.1 in main text). In the LiDAR branch, for ppScore calculation [10], we set the radius for counting neighboring points at 0.3 meters. After ppScore calculation, the RANSAC [3] ground removal algorithm is applied to every LiDAR point cloud. For graph construction, we set the distance threshold r_t at 2 meters. For graph-based clustering, we use DBSCAN [2] and set parameter ϵ at 0.1 and the $sample_{min}$ at 10. We convert nuScenes [1] into KITTI [4] format and primarily consider the front view, where only pseudo boxes generated in front of the ego car are saved. In the image branch, GroundingDINO [7] with the SwinB backbone serves as our 2D detector. We choose a box score threshold of 0.25 and a text score threshold of 0.24. The text prompt adopted is "car . truck . trailer . bus . bicycle . motorcycle . pedestrian . cone . barrier . construction vehicle ." Boxes classified as "cone" and "barrier" are filtered out at the end of the generation process. The version of Segment-Anything-Model (SAM) [5] is vit_h. For 3D boxes integration, image-based 3D boxes are filtered based on their depth, which indicates the distance from the ego car.

Data Processing. For data augmentation, we utilize a random world flip along the x-axis, a random world rotation with an angle range of (-0.785, 0.785), and a random world scaling with a scale range of (0.95, 1.05). For data processing, we apply point sampling to reduce every point cloud to 6144 points in both the train and test sets. Point shuffling is employed in the train set, but not during testing. We utilize voxelization for point clouds with voxel sizes of (0.05, 0.05, 0.1), a maximum of 5 points per voxel, and a cap of 16000 voxels on the train set and 40000 on the test set. In the nuScenes detection task, which includes

* Corresponding author.

10 classes, we focus on dynamic classes: bicycle, bus, car, construction vehicle, motorcycle, pedestrian, trailer, and truck, excluding static classes like barrier and cone. The corresponding camera images have a resolution of 1600×900. Although nuScenes [1] provides nine sweeps of LiDAR point clouds following each key sample, our experiments solely harness key sample point clouds without incorporating the sweeps to keep a fair comparison with existing work [10].

Backbone. All experiments are conducted using PointRCNN [9], which adopts PointNet2 [8] as the backbone for 3D feature extraction. PointRCNN contains two separate heads for 3D box localization and classification, respectively. Specifically, it includes four set abstraction layers, with point group sizes of 4096, 1024, 256, and 64, along with 4 feature propagation layers. The classification head employs sigmoid focal classification loss and the regression head adopts weighted smooth l1 loss.

Training Details. The model is trained for 11 rounds, including one seed training round and 10 self-paced training rounds. In each training round, the model undergoes training for 80 epochs, with checkpoints saved every 10 epochs. Each training round takes approximately 3 hours, culminating in a total runtime of around 33 hours.

A.2 More Qualitative Results

In Figure 1, we provide additional visualizations illustrating the enhanced performance of our model on distant and small objects. These visualizations further validate the effectiveness of our proposed integration with 2D scenes, as well as the adaptive sampling strategy and weak model aggregation. These components collectively enhance the detection ability on samples which are challenging for LiDAR-based methods.

A.3 Analysis

Motivation for Distance-Aware Integration (see Section 3.1 in main text). Considering the LiDAR point density at the close range, LiDAR-based methods tend to detect nearby objects easily. Meanwhile, image-based methods are also proficient at identifying objects in the near range since they exhibit evident shape and texture features. Direct integration can lead to excessive and overlapped pseudo-box estimation for nearby objects. We present the outcomes of directly integrating image-based pseudo-boxes with LiDAR-based pseudo-boxes in Figure 2. Such an integration strategy raises potential conflicts between different pseudo boxes and can degrade the final detection performance of the model.

We thus propose a more advanced distance-aware integration method and filter out image-based pseudo boxes at close range. More specifically, we conduct extensive ablation studies on various ranges, such as $> 5m$, $> 10m$, and $> 15m$, and find that integrating image-based 3D boxes in range $> 10m$ into LiDAR-based 3D boxes is optimal. It avoids near-range conflicts and maximizes the utilization of image-based pseudo boxes.

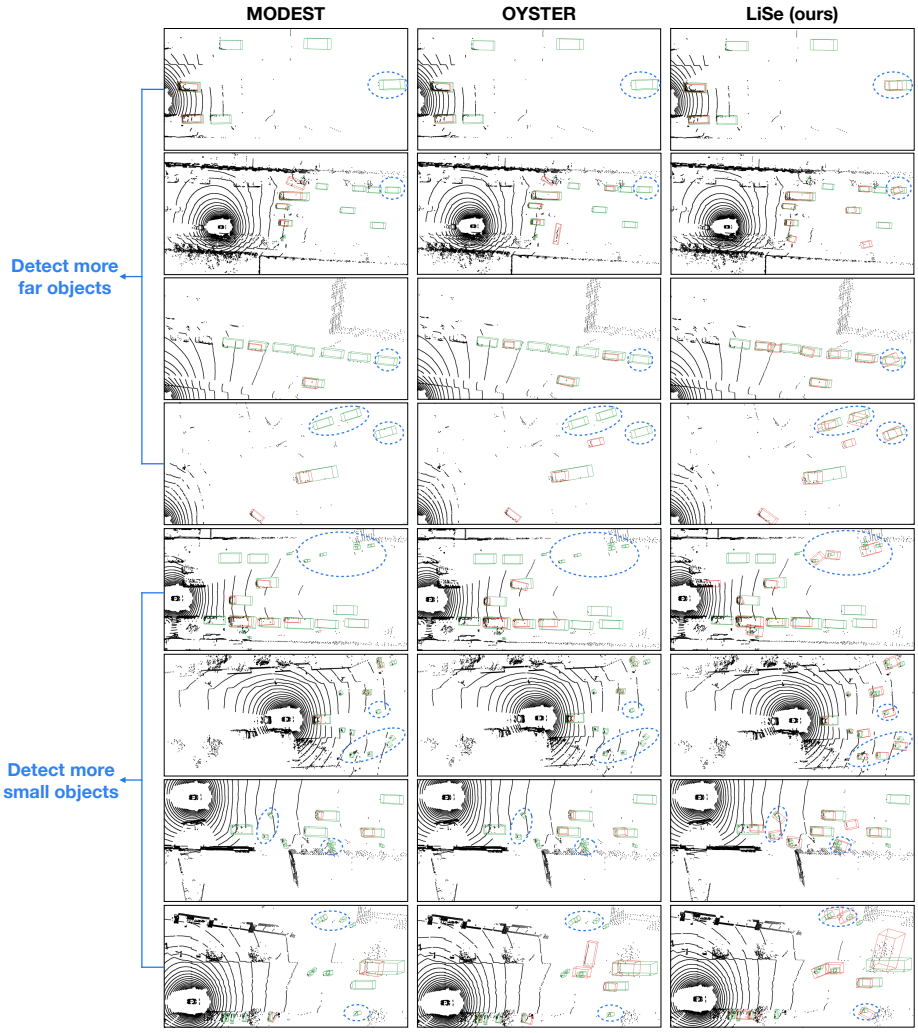

Fig. 1: Visualization comparison between MODEST [10], OYSTER [11], LiSe (ours), and ground truth boxes. All results are from the best-performing models. We show results from 8 different locations and each row represents one location. The overall results indicate that our model is superior in detecting distant and small objects. Best viewed in color: green boxes represent ground truth labels, red boxes indicate predictions, and blue circles highlight differences in predictions.

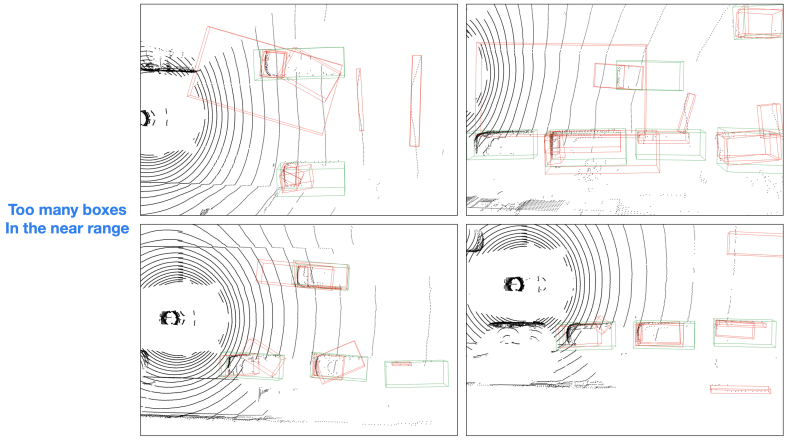

Fig. 2: This figure illustrates the issue arising from simply combining LiDAR-based and image-based pseudo boxes. Both generation methods effectively identify objects at close range, leading to box redundancy in this area, which often results in conflicts. Such conflicts can negatively impact model performance. The boxes are best viewed in color: green boxes represent ground truth labels and red boxes indicate generated pseudo boxes.

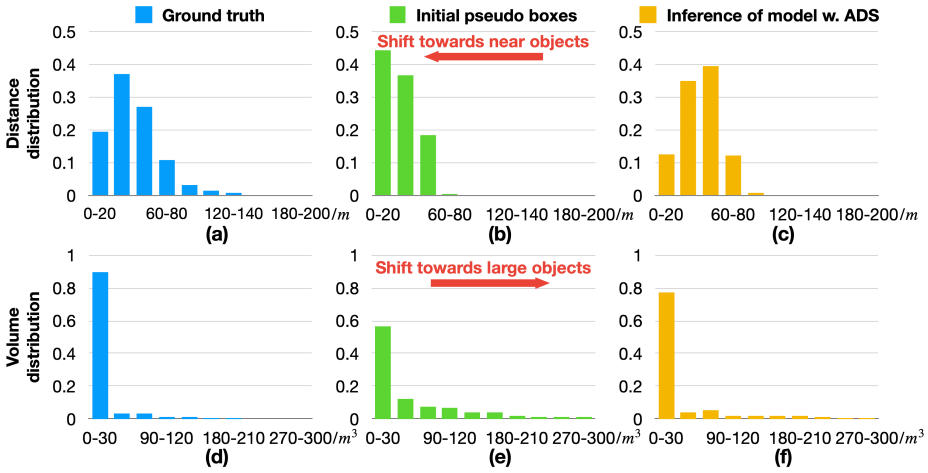

Fig. 3: This figure illustrates the motivation behind our adaptive sampling strategy (ADS). Panels (a), (b), and (c) depict the distance distributions of ground truth boxes, initial pseudo boxes of first self-paced learning round, and inference results of model trained with our adaptive sampling strategy, respectively. Panels (d), (e), and (f) correspond to the volume distributions. A significant distance shift towards near objects is noticeable when comparing (a) and (b), and a similar volume shift towards large objects is evident in (d) and (e). Our adaptive sampling strategy effectively mitigates these shifts, as shown in (c) and (f).

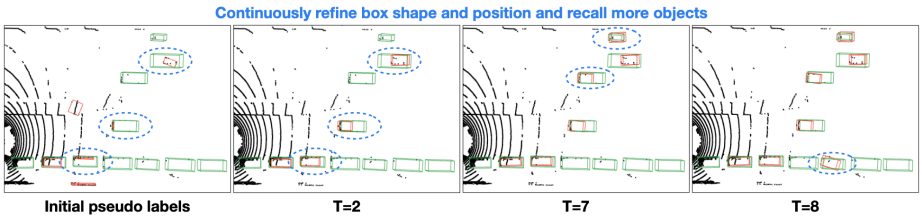

Fig. 4: Visualization of pseudo boxes and detection results of self-paced learning process. LiSe can keep refining box shape and position and recall more objects.

Motivation for Adaptive Sampling Strategy (see Section 3.2 in main text). We evaluate the distribution of boxes based on unique 3D world attributes such as distance and volume and identify a pronounced bias towards simpler samples, such as near and large objects. Such a phenomenon is illustrated in Figure 3. To address these biases, we devise the adaptive sampling strategy, which dynamically adjusts the sampling rate for different object groups based on feedback from model through a self-paced learning process [6]. As depicted in Figure 3, the adaptive sampling strategy effectively counterbalances these biases. The efficacy of this approach is further validated by our experiment results.

A.4 Discussion

Why are the initial results of LiSe (T=0) lower than MODEST [10] and OYSTER [11] (see Table 1 in main text)? For LiSe, we generate pseudo boxes by integrating data from both LiDAR and image modalities. This integration brings the noise from both modalities, potentially lowering the accuracy of the boxes compared to those solely based on LiDAR. On the other hand, image-based pseudo boxes complement LiDAR-based ones, particularly for distant and small objects, enhancing the overall diversity of the pseudo labels. This diversity explains why LiSe initially lags behind MODEST and OYSTER but continuously improves in subsequent training rounds, while MODEST and OYSTER peak early and then degrade. Ultimately, LiSe significantly outperforms these models.

Why does not experiment consider static classes, such as "Cone" and "Barrier"? The competitive LiDAR-based methods, *e.g.*, MODEST [10], can only detect dynamic objects according to the position movements. In the experiments, we exclude static object classes like cone and barrier for a fair comparison. It is worth noting that our proposed integration with 2D scenes can enable the detection of static objects by adding image-based pseudo boxes of static classes into the initial training labels.

A.5 Pseudo boxes quality

Initial pseudo boxes are not very accurate. Our method can continuously refine the shape and location of its detection and recall more objects during self-

training process (see Figure 4). In the end, our model predictions are much more accurate than the initial labels.

Acknowledgement

The paper is supported by Start-up Research Grant at the University of Macau (SRG2024-00002-FST).

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
2. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
5. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
6. Kumar, M., Packer, B., Koller, D.: Self-paced learning for latent variable models. Advances in neural information processing systems **23** (2010)
7. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
8. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
9. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
10. You, Y., Luo, K., Phoo, C.P., Chao, W.L., Sun, W., Hariharan, B., Campbell, M., Weinberger, K.Q.: Learning to detect mobile objects from lidar scans without labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1130–1140 (2022)
11. Zhang, L., Yang, A.J., Xiong, Y., Casas, S., Yang, B., Ren, M., Urtasun, R.: Towards unsupervised object detection from lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9317–9328 (2023)