Supplementary Material for Revisiting Domain-Adaptive 3D Object Detection by Reliable, Diverse and Class-balanced Pseudo-Labeling

Zhuoxiao Chen[1] Yadan Luo[1] Zheng Wang[2] Mahsa Baktashmotlagh[1] Zi Huang[1]
[1]The University of Queensland [2]University of Electronic Science and Technology of China
{zhuoxiao.chen, y.luo, m.baktashmotlagh, helen.huang}@uq.edu.au, zh_wang@hotmail.com

In this supplementary material, we provide a detailed description of the proposed algorithm (Sec 1), a conceptual comparison with previous method (Sec 2) and additional experimental results and analysis (Sec 3).

1. Algorithm

To thoroughly describe the procedure of unsupervised domain-adaptive 3D object detection by the proposed ReDB, we present the Algorithm 1. In the stage one, we pretrain the 3D detector $F(\cdot)$ on the source domain with the randomly scaled objects [6] $\{(X_i^s, f_{\text{ROS}}(Y_i^s))\}_{i=1}^{N_s}$. If the current epoch is initial $e = 1$ or in the list $L = [31, 61, 91]$, which specifies the epochs requiring pseudo-labelling, we enter stage 2 to generate reliable, diverse and balanced ReDB pseudo labels. We first inference the target domain $\{X_i^t\}_{i=1}^{N_t}$ and obtain the pseudo label $\{\widehat{Y}_i^t\}_{i=1}^{N_t}$ for each target point cloud. Then, if the current epoch $e = 1$, which means the 3D detector has no knowledge about the target domain, the pseudo label might be unreliable thus we need examination. We feed the pseudo label paired with the corresponding point clouds $\{(X_i^t, \widehat{Y}_i^t)\}_{i=1}^{N_t}$ and the source pairs $\{(X_i^s, Y_i^s)\}_{i=1}^{N_s}$ into the proposed Cross-domain Consistency Examination (CDE) module that filters out the unreliable pseudo labels via Equation (2), (3). Once the reliability of pseudo labels is guaranteed, we target at obtaining a more geometrically diverse subset of pseudo-labeled boxes $\{\hat{b}_j\}_{i=j}^{\hat{B}/d}$ to close the object gap from different geometrical aspects, using OBC-based downsampling via Equation (4), (5). Next, to alleviate the class imbalance, we sample class-balanced reliable and diverse pseudo labels $\{\hat{b}_j\}_{j=1}^{\hat{B}/d}$ and randomly scaled source labels $f_{\text{ROS}}(\{b_j\}_{j=1}^{B})$ to each point cloud $\{X_i^t, \widehat{Y}_i^t\}_{i=1}^{N_t}$ via Equation (6), (7). At the end of stage 2, we have a subset of ReDB pseudo labels. We train this subset in stage 3 for several epochs until the current epoch appears in the pseudo-labelling epoch list L. The algorithm is then alternating between stage 2 and stage 3 until running out of all epochs.

2. Conceptual Comparison

This section establishes a connection between the proposed ReDB and previous domain-adaptive 3D detection approaches, with emphasis on real-world applicability and aspects to address domain shifts, as summarized in Tab 1.

Statistical Normalization (SN) [4]: A data modification approach that modifies the object size of the source domain to match the target statistics, in order to address scale-induced object shift. Nonetheless, this method demands the access to object statistics of the target domain, rendering it unfeasible in practical scenarios where domain knowledge is not available.

MLC-Net [2]: A teacher-student paradigm, in which the teacher parameters are aggregated by an exponential moving average (EMA) of the student model and updated iteratively. The teacher is in charge of producing pseudo labels that supervise the learning of the student model. The technique leverages the historical weights to predict smooth pseudo labels for self-training on the target domain. However, this method suffers from four drawbacks: (1) it overlooks the environmental gap, leading to erroneous pseudo-labeling during the initial pseudo-label generation stage. However, our proposed cross-domain examination (CDE) offers a simple solution to this problem. (2) The employment of two models (student and teacher) simultaneously doubles the training time and GPU memory consumption. Additionally, the point-wise consistency loss that supervised the learning of student model yields more gradient back-propagation, which further increases the training time. (3) Despite being compatible with point-based 3D detectors, this method cannot be applied to mainstream grid-based detectors. (4) This approach only tailors the model training and inference for a single class, which is not a viable strategy in real-world situations.

LiDAR Distil [5]: A teacher-student framework that seeks to enhance the generalizability of 3D detectors to the domain with different beam numbers. In particular, the teacher and student networks are trained using high-beam

Algorithm 1 The algorithm of ReDB for domain-adaptive 3D object detection

```
Input:
  $\{(X_i^s, Y_i^s)\}_{i=1}^{N_s}$ - source point clouds with human annotations, where all labelled boxes: $\{b_j\}_{j=1}^{B} = \{b_j | b_j \in Y_i^s\}_{i=1}^{N_s}$
  $\{X_i^t\}_{i=1}^{N_t}$ - target point clouds without human annotations
  $F(\cdot)$ - the 3D object detector
  $f_{\text{ROS}}(\cdot)$ - the random object scaling funcion
  $E$ - total number of training epochs
  $L$ - a list of epochs requiring pseudo labelling
Output:
  $F(\cdot)$ the 3D detector adapted to the target domain

  /* Stage 1: Pretrain on the Source Domain */
  $F(\cdot) \leftarrow \{(X_i^s, f_{\text{ROS}}(Y_i^s))\}_{i=1}^{N_s}$
  for $e \in [1, \cdots, E]$ do
      if $e = 1$ or $e \in L$ then
          /* Stage 2: Generate Pseudo Labels on the Target Domain */
          $\{\widehat{Y}_i^t\}_{i=1}^{N_t} \leftarrow F(\{X_i^t\}_{i=1}^{N_t})$
          if $e = 1$ then                          ▷ Cross-domain Consistency Examination via Equation (2), (3)
              $\{\widehat{Y}_i^t\}_{i=1}^{N_t} \leftarrow \text{Reliability}(\{(X_i^t, \widehat{Y}_i^t)\}_{i=1}^{N_t}, \{(X_i^s, Y_i^s)\}_{i=1}^{N_s})$
          end if
          $\{\hat{b}_j\}_{j=1}^{\hat{B}} = \{\hat{b}_j | \hat{b}_j \in \hat{Y}_i^t\}_{i=1}^{N_t}$
          $\{\hat{b}_j\}_{j=1}^{\hat{B}/d} \leftarrow \text{Diversity}(\{\hat{b}_j\}_{j=1}^{\hat{B}})$          ▷ OBC-based Downsampling via Equation (4), (5)
          $\{(X_i^t, \widehat{Y}_i^t)\}_{i=1}^{N_t} \leftarrow \text{Balance}(\{(X_i^t, \widehat{Y}_i^t)\}_{i=1}^{N_t}, \{\hat{b}_j\}_{j=1}^{\hat{B}/d}, f_{\text{ROS}}(\{b_j\}_{j=1}^{B}))$
                                                   ▷ Class-balanced Sampling via Equation (6), (7), (8)
      end if
      /* Stage 3: Self-train on the Target Domain */
      $F(\cdot) \leftarrow \{(X_i^t, \widehat{Y}_i^t)\}_{i=1}^{N_t}$
  end for
```

Table 1: Comparison of prior Domain Adaptive 3D Detection methods in two aspects: applicability and addressed domain shifts. **The main difference is that the existing methods has constrained applicability, and address incomprehensive domain shifts.**

	Applicability				Addressed Domain Shift	
Method	Multi-class	Compatible	UDA	Space Complexity	Object Shift	Environmental Shift
SN [4]	✔	✔	✘	✔	Scale	✘
LiDAR Distil [5]	✘	Grid-based	✘	Multiple models	✘	Beam number
MLC-Net [2]	✘	Point-based	✔	Dual models	Scale	✘
ST3D [7]	✘	✔	✔	Memory bank	Scale	✘
ST3D++ [6]	✘	✔	✔	Memory bank	Scale	✘
ReDB	✔	✔	✔	✔	Diversity geometrics (scale, density, distance, etc)	All aspects (beam number, angle, etc)

and low-beam data, respectively. The student network is subsequently trained to align the regions of interest on the bird eye vew (BEV) feature maps with those of the teacher model. The primary objective of this approach is to mitigate the environmental gap that stems from inconsistent beam numbers. However, this method encounters similar shortcomings as MLC-Net or SN, including the prerequisite knowledge about the target domain (*i.e.*, beam number), the employment of multiple teacher networks leading to considerable computation time and GPU memory consumption, restricted applicability to point-based 3D detectors, and unfair single-class adaptation. Additionally, this study fails to account for shifts in objects and other environmental factors, such as disparities in beam angle, range, and data collection locations and time.

ST3D [6] / ST3D++ [7]: A self-training strategy that em-

S^r	S^g	$S^\triangle$	mAP$_{\text{BEV}}$ / mAP$_{\text{3D}}$
0	10	0	56.73 / 45.21
0	10	2	60.27 / 49.25
10	20	2	60.22 / 48.73
5	10	2	**61.14 / 50.10**

Table 2: Sensitivity analysis for S^r, S^g and $S^\triangle$.

ploys random object scaling (ROS) to pre-train the source data and a memory bank to store and update all pseudo labels for self-training the target data. The objective of ROS is to alleviate object shift by allowing the pre-trained detector to recognize objects with a wide range of scales. The memory bank combines historical and current pseudo labels to generate more consistent pseudo labels. ST3D++ is an extended version that incorporates a domain-specific batch normalization [1] with the aim of disentangling the statistic estimation (*i.e.*, mean and variance) in different domains within batch normalization layers. The benefit of this series of work (*i.e.*, ST3D and ST3D++) over prior works is that, they do not necessitate any prior knowledge of the target domain and are not constrained by detector types. Despite some progress, significant challenges persist in several aspects, including neglecting environmental shifts, unfair single-class adaptation, and excessive storage requirements for retaining historical pseudo labels.

REDB: Tab 1 highlights the advantages offered by the proposed REDB, concerning the approach applicability and handled domain shifts. The proposed REDB (1) facilitates multi-class adaptation via the proposed class-balanced self-training, (2) is compatible with all types of point cloud encoders used in modern 3D detectors, (3) employs an unsupervised domain adaptation (UDA) approach, which eliminates the need for any prior knowledge about the target domain, (4) does not incur extra costs for GPU memory and disk storage, (5) lastly, provides a solution to comprehensively identify and address both object shifts and environmental shifts.

3. Additional Experiments

3.1. Implementation Details

Hyperparameter settings. For self-training on the target domain, we set the total training epochs E as 120 and pseudo label generations list L as $[31, 61, 91]$. Thus, we generate the pseudo labels at the initial epoch of self-training, then update pseudo labels every 30 training epochs. **We provide complete configuration files for all experiments in our code repository, attached with the supplementary material.**

Fair Model Selection. The prior approaches evaluate the checkpoints based on the performance of the target domain, which is **not fair and impractical**, because the target labels not observable under the unsupervised domain adaptation (UDA) setting. Hence, We again revisit model selection in a fair manner without accessing any target labels. For selecting the pre-trained models, we simply opt for the one with the **lowest source risk**. Regarding the self-trained checkpoints, we observe that the modern 3D detectors optimized with Adam Onecycle [3] are typically reaching the best performance at the cycle's end. Based on this observation, we conduct multiple training cycles (each cycle contains 30 epochs) with the Adam Onecycle optimizer and generate pseudo-labels at the end of each cycle. The final selected model is the **last checkpoint of the last round** after several rounds of self-training. By doing so, we not only assure the pseudo-labels are generated by optimal models at each round, but also can simply decide the last checkpoint as the final model without knowing the target labels.

Method	nuScenes → KITTI	Waymo → nuScenes
ST3D	25h 24m 48s	27h 13m 42s
ST3D++	22h 22m 28s	**24h 57m 23s**
REDB	**20h 21m 1s**	29h 53m 16s

Table 3: Self-training Time Comparison.

3.2. Sensitivity Analysis of S^r, S^g and $S^\triangle$

In this section, We examine the sensitivity of our approach to different values of hyperparameters S^r, S^g and the increasing and decreasing number $S^\triangle$ for S^r and S^g, respectively. Tab 2 presents different combinations of S^r, S^g and $S^\triangle$. A comparison of the row-1 and row-2 reveals that injecting pseudo-labeled REDB objects progressively, can yield a notable performance boost, which are 6.24% and 8.94% in mAP$_{\text{BEV}}$ and mAP$_{\text{3D}}$, respectively. When comparing row-2, row-3 and row-4, we find that either sampling a large or small number of objects leads to similar performance, fluctuating at 0.87% mAP$_{\text{BEV}}$ and 1.37 mAP$_{\text{3D}}$, respectively. As a result, to secure an appropriate time complexity, we commonly select the values in row-4 for S^r, S^g and $S^\triangle$.

3.3. Statistical Analysis on OBC scores

In this section, we plot the statistical correlation of the proposed overlapped box counting (OBC) with different geometrical features in Fig 1. The first plot shows that there is a clear relationship between OBC and box scales, resulting in a Pearson correlation coefficient (PCC) of 0.64. The second and third plots suggest that point density and object-to-sensor distance have a minor correlation with the OBC score, with PPC 0.21 and 0.33, respectively. Such minor correlations are discovered because the diversity of an object is not solely determined by a single factor (*e.g.,* object density, distance), but an unmeasurable combination of different geometrical features. To address multi-factor object shifts in 3D scenes, a universal metric is required. Instead

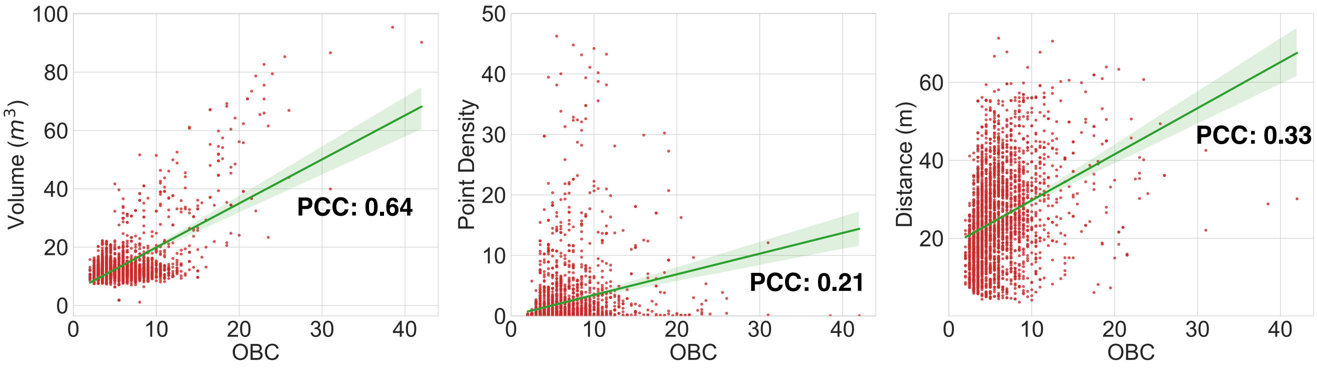

Figure 1: Statistical analysis on overlapped boxes counting (OBC). Each red point indicates a pseudo-labeled box associated with its OBC value and 1-dimensional geometrical feature (*e.g.*, volume, point density and distance). The green line is the regression line, and the translucent band indicates the size of the confidence interval for the regression estimate. We calculate the Pearson correlation coefficient (PCC) to show the correlation between OBC value and different geometrical features.

of manually designing a linear combination of different factors, the proposed OBC score focuses on reflecting the overall diversity for each 3D object.

3.4. Time Analysis on Self-training

To demonstrate that the effectiveness of the proposed REDB does not benefit from additional extensive computation, we report the runtime of the entire self-training process, including pseudo label generation, in Tab 3. Note that for all compared approaches, we use two Tesla V100-PCIE-16GB GPUs for the nuScenes → KITTI task and one NVIDIA RTX A6000 GPU for the Waymo → nuScenes task. The backbone 3D detector is SECOND for all approaches. The time comparison results show that our method takes a similar amount of time as the two existing baseline methods. Specifically, on the task of nuScenes → KITTI, we are approximately 2 to 4 hours faster, while 3 to 5 hours slower for the Waymo → nuScenes task. This is due to the fact that each single frame of point cloud in Waymo is much larger than that in nuScenes, thus CDE takes longer to infer the pseudo labels pasted to the Waymo point cloud. However, both existing methods rely on the memory bank technique, which is not only time-consuming but also memory-hungry.

References

[1] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7354–7362, 2019. 3

[2] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proc. International Conference on Computer Vision (ICCV)*, pages 8846–8855, 2021. 1, 2

[3] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *CoRR*, abs/1803.09820, 2018. 3

[4] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark E. Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the USA: making 3d object detectors generalize. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11720, 2020. 1, 2

[5] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *Proc. European Conference on Computer Vision (ECCV)*, volume 13699 of *Lecture Notes in Computer Science*, pages 179–195, 2022. 1, 2

[6] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: self-training for unsupervised domain adaptation on 3d object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10368–10378, 2021. 1, 2

[7] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2