A-Teacher: Asymmetric Network for 3D Semi-Supervised Object Detection

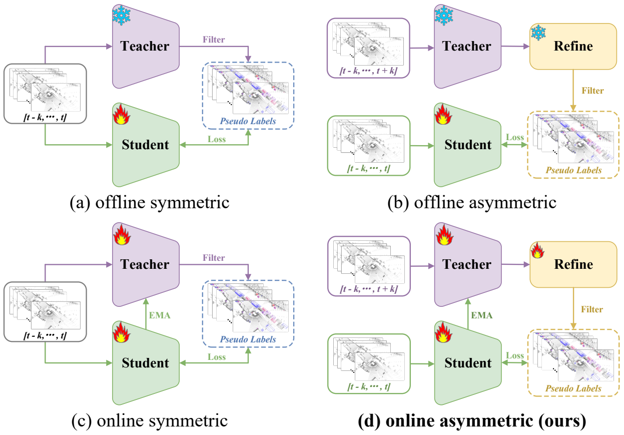
Hanshi Wang[1,2], Zhipeng Zhang[3*], Jin Gao[1,2*], Weiming Hu[1,2,4]
[1]State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), CASIA
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]KargoBot
[4]School of Information Science and Technology, ShanghaiTech University
{hanshi.wang.cv, zhipeng.zhang.cv}@outlook.com, {wmhu, jin.gao}@nlpr.ia.ac.cn

Abstract

This work proposes the first ***online asymmetric*** *semi-supervised framework, namely A-Teacher, for LiDAR-based 3D object detection. Our motivation stems from the observation that* ***1)*** *existing* ***symmetric*** *teacher-student methods for semi-supervised 3D object detection have characterized simplicity, but impede the distillation performance between teacher and student because of the demand for an identical model structure and input data format.* ***2)*** *The* ***offline asymmetric*** *methods with a complex teacher model, constructed differently, can generate more precise pseudo labels, but is challenging to jointly optimize the teacher and student model. Consequently, in this paper, we devise a different path from the conventional paradigm, which can harness the capacity of a strong teacher while preserving the advantages of jointly updating the whole framework. The essence is the proposed attention-based refinement model that can be seamlessly integrated into a vanilla teacher. The refinement model works in the divide-and-conquer manner that respectively handles three challenging scenarios including 1) objects detected in the current timestamp but with sub-optimal box quality, 2) objects are missed in the current timestamp but are detected in supporting frames, 3) objects are neglected in all frames. It is worth noting that even while tackling these complex cases, our model retains the efficiency of the online semi-supervised framework. Experimental results on Waymo [38] show that our method outperforms previous state-of-the-art HSSDA [17] for 4.7 on mAP (L1) while consuming fewer training resources.*

1. Introduction

Recent years have witnessed the rapid development of autonomous driving. The perception algorithm is the starting point of the whole system. 3D object detection with point clouds is the predominant part of many perception systems attributed to the robustness of LiDAR sensors. A plethora of models have been proposed to improve the efficiency [5, 13, 15, 25, 33, 45] and effectiveness [10, 16, 30, 50, 51, 57] of point cloud detection methods. Despite evaluation metrics like mAP [2] being continuously improved, it is still challenging to handle all cases in real-world scenarios. This naturally leads to the question: *what is the next potential path for evolving LiDAR-based object detection?*



(a) offline symmetric

(b) offline asymmetric

(c) online symmetric

(d) online asymmetric (ours)

Figure 1. Illustration of the teacher-student semi-supervised framework: a) offline symmetric paradigm, b) offline asymmetric paradigm, c) online symmetric paradigm, **d**) the proposed online asymmetric design.

Recent strides in artificial general intelligence (AGI) like ChatGPT [23] and SAM [11] prove that *data-driven with semi-supervised learning* could be a promising answer. Upon revisiting the developing routes in LiDAR-based 3D object detection, we also note a burgeoning interest in semi-supervised approaches within the research community. Among them, the teacher-student framework emerges as the predominant branch, and has developed three main streams. As illustrated in Fig. 1, based on whether the teacher model is frozen, the semi-supervise approaches are divided into **offline**, *i.e.,* (a) and (b), and **online** methods

[*] Corresponding author.

i.e., (c). The offline one only annotates the unlabeled data a few times in the whole training process [3], discarding online updating of the teacher model. Therefore, a powerful but complex teacher with a refinement model is recently exploited to improve the quality of pseudo labels [9, 19] (Fig. 1(b)). This provides another taxonomy and forms **symmetric** and **asymmetric** architectures, *i.e.* (a) $v.s.$ (b).

Delving into the three existing frameworks, it is observed that each of them has pros and cons. The offline symmetric method, *i.e.,* Fig. 1(a), is characterized by its simplicity. However, its effectiveness is constrained, primarily due to the limited capacity of the teacher model which is restricted to the information gleaned from the small amount of labeled data. The online symmetric method, depicted in Fig.1(c), represents an enhancement over the offline symmetric counterpart Fig. 1(a) by updating the teacher model. This adaptive approach enables the teacher to assimilate knowledge from unlabeled data [17, 24, 37, 40]. However, to maintain training efficiency and meet the demand of EMA [39], the complexity of the teacher model is constrained, which generally adopts the same architecture as the student. Another different path is the offline asymmetric method, *i.e.,* Fig. 1(b), which leverages future frames to augment the performance of the current timestamp with an auxiliary refinement model [9, 19, 29, 48]. To ensure the efficacy of the refinement model, a powerful but complex multi-object tracker (MOT) is typically used to refine the box sequences. Therefore, it is unlikely to jointly train the student, teacher, and refinement models. A question is: *Can we harness the temporal refinement benefits from asymmetric architecture while simultaneously facilitating online updates to enable the teacher to absorb knowledge from unlabeled data?*

We show the answer is affirmative by proposing an online asymmetric semi-supervised framework for LiDAR-based object detection, which seeks to capture the temporal refinement benefits of offline asymmetric paradigm while incorporating continuous teacher updates of online symmetric designs. As previously discussed, the barrier to achieving this goal is the lack of a light refinement model. Hence in this work, we propose an efficient attention-based refinement model to aggregate temporal information. The refinement model consists of three essential components, each enhancing the quality of pseudo labels from different perspectives. Specifically, the propagation-based aggregation module refines successfully detected objects in the current timestamp by propagating the boxes to supporting frames and then merging box-sequence information. Conversely, the dreaming-based box aggregation tries to recall the false negatives by clustering boxes detected in the supporting frames and verifying their presence in the current timestamp. Last but not least, for objects missed in both current and supporting frames, we introduce spatio-temporal deformable aggregation to fuse features from different frames. The enhanced representations contribute to reducing false negatives. Importantly, our proposed refinement model discards the need for complex tracking designs, enabling joint training with the teacher-student network. By applying the proposed framework to PV-RCNN [36], the model achieves improvements of 18.9 L1 mAP on Waymo [38] benchmark compared with the supervised baseline.

In summary, our main contributions are as follows: **1)** We propose an attention-based lightweight refinement model to improve the quality of pseudo labels generated by the teacher model; **2)** We design an online asymmetric semi-supervised framework for LiDAR-based object detection, which can simultaneously absorb the temporal refinement benefits of offline asymmetric paradigm and contiguous joint update in online design; **3)** We conduct extensive experiments to demonstrate the effectiveness of our framework on different single- or multi-frame based methods, including PV-RCNN [35], Second [47], and Voxel-RCNN [7].

2. Related Work

2.1. LiDAR-based 3D Object Detection

LiDAR-based object detection plays an essential role in autonomous driving. Recent years have witnessed rapid development in this field, and the research interest can be roughly divided into point-based methods [27, 28, 49, 55], voxel-based methods [22, 32, 52, 54], and hybrid one that combines both of them [7, 20]. In particular, PointNet[26] and PointRCNN[34] directly extract features from the raw point clouds to effectively retain the geometric information. VoxelNet[59] convert point clouds into voxels and then efficiently process the voxels with convolution layers [47]. PV-RCNN [35] and PV-RCNN++ [36] combine the advantages of the two streams, which can simultaneously guarantee computational efficiency and flexible receptive field. Another stand-alone interest in LiDAR-based object detection is leveraging temporal information from continuous sensor recording, which benefits location robustness and speed estimation. DSVT [41] and SST [8] simply concatenate point clouds of different timestamps. CenterFormer[60] uses transformer layers to align and fuse bird's eye view (BEV) features from different frames. MPPNet [4] uses an online tracker to link objects in the temporal dimension.

2.2. Semi-Supervised Object Detection

Semi-supervised learning, which aims to absorb nutrients from both labeled and unlabeled data, has achieved astonishing improvements in recent years. 3D object detection, especially with LiDAR sensors, benefits a lot from the rapid development of semi-supervised learning, since it is impossible to annotate extremely massive data from vehicles. Semi-supervised methods thus become economical choices. The related works can be categorized into two categories,

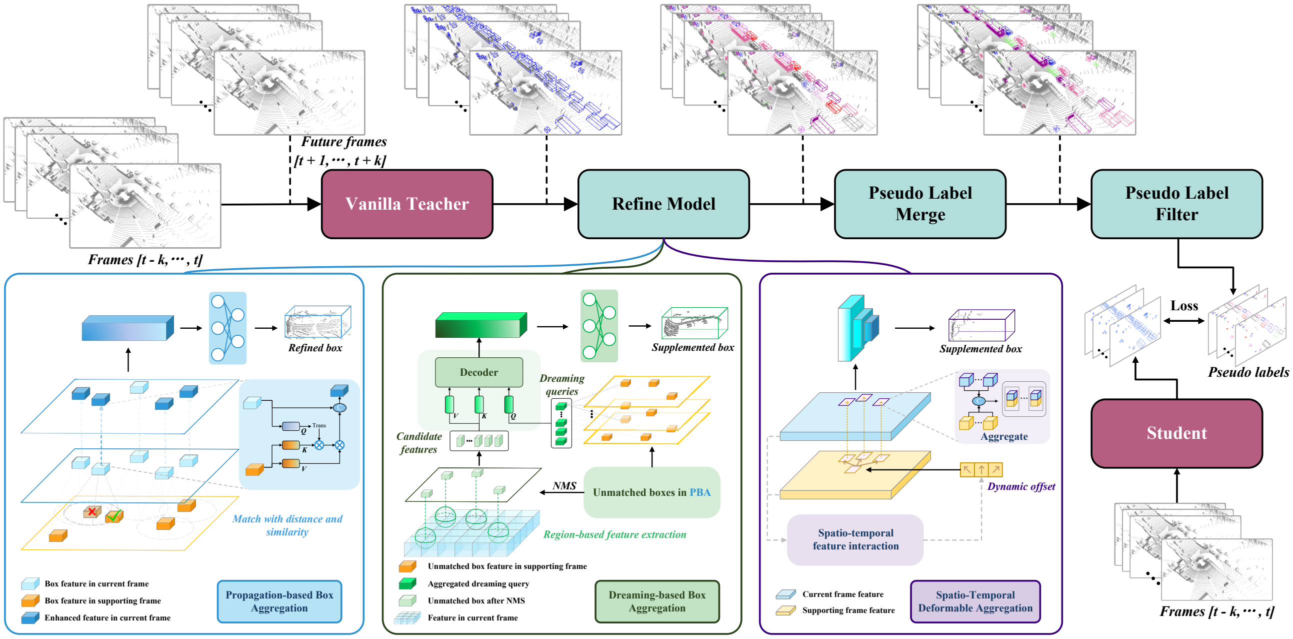


Figure 2. **Overview of the proposed online asymmetric semi-supervised framework.** (1) The vanilla teacher generates the candidate boxes of $t-k, \cdots, t+k$ frames. (2) The efficient refinement model that includes **PBA**, **DBA** and **STA** is proposed to generate pseudo labels. (3) Merge the pseudo labels from each refinement component and filter the low-quality pseudo labels. (4) The student model is supervised by the pseudo labels generated by the teacher and refinement model. (5) The teacher is updated with student's weights by EMA.

where the consistency-based methods [18, 46, 58] apply different perturbations and augmentations to the input data, and then optimize the model by minimizing the consistency loss, and the pseudo-label based methods [1, 14, 37, 43, 44] first train a teacher with the labeled dataset, then a student model is trained on the unlabeled dataset which has pseudo annotated by the teacher.

In particular, SESS [56] designs a consistency loss to optimize the model with the input data of different augmentations. 3DIoUMatch [37, 40] proposes a filtering strategy to improve performance by filtering out low-quality pseudo labels. DetMatch [24] simultaneously predicts 2D and 3D bounding boxes on the unlabeled images and point clouds, showing that 2D visual tasks also help 3D point cloud object detection. Proficient Teachers [53] first applies data augmentation to generate multiple point clouds and then uses teachers from different training periods to predict pseudo boxes, which improves the recall of pseudo label generation. HSSDA [17] proposes a hierarchical threshold learning strategy to separate the pseudo labels for different tasks. NoiseDet [6] builds the noise-resistant instance supervision module and pixel-wise feature consistency constraints to generate and purify pseudo labels. In this work, we for the first time propose an online asymmetric semi-supervised framework for LiDAR-based object detection.

3. Methodology

3.1. Preliminary and Overview

Our model employs the teacher-student paradigm to construct a LiDAR-based semi-supervised learning framework. Given the labeled data $\mathbf{D}_\mathrm{l} = \{\mathbf{x}_\mathrm{i}^\mathrm{l}, \mathbf{y}_\mathrm{i}^\mathrm{l}\}_{\mathrm{i}=1}^{\mathbf{N}_\mathrm{l}}$ and unlabeled data $\mathbf{D}_\mathrm{u} = \{\mathbf{x}_\mathrm{i}^\mathrm{u}\}_{\mathrm{i}=1}^{\mathbf{N}_\mathrm{u}}$ $(N_\mathrm{u} \gg N_\mathrm{l})$, the teacher model is first trained on $\mathbf{D}_\mathrm{l}$ to generate pseudo labels $\mathbf{y}_\mathrm{i}^\mathrm{u}$ on $\mathbf{D}_\mathrm{u}$, and then the student model learns knowledge from both $\mathbf{D}_\mathrm{l}$ and $\mathbf{D}_\mathrm{u}$. During training, the teacher model is iteratively updated with exponential moving average (EMA) [39],

$$\theta_{\mathrm{tea}}^{\mathrm{s}} \leftarrow \alpha\theta_{\mathrm{tea}}^{\mathrm{s}-1} + (1-\alpha)\theta_{\mathrm{stu}}^{\mathrm{s}}, \tag{1}$$

where θ_{tea} and θ_{stu} represent the parameters of the teacher and student models, respectively. α is a smoothing coefficient, s is the training step. In the context of LiDAR-based 3D object detection, the input data $\mathbf{x}$ is point cloud scans, and the label $\mathbf{y}$ includes location, size, heading angle (yaw), and category information of objects.

As shown in Fig 2, we propose an online asymmetric framework, where the teacher adopts an attention-based refinement model to summarize information from past and future timestamps (supporting frames) with the goal of improving the quality of the pseudo label on the current frame. Specifically, our semi-supervised framework contains four essential steps: 1) Firstly, the vanilla teacher, which is a single-frame detector, predicts bounding boxes for all input frames ranging from $t-k$ to $t+k$. 2) Secondly, the proposed attention-based refinement model, detailed in Sec. 3.2, aggregates past and future information to refine the detection results from the vanilla single-frame detector. 3) Thirdly, a dual-threshold strategy is introduced to filter low-quality proposals and generate the pseudo labels for training the student model, which is further elaborated in Sec.3.3. 4) Lastly, we update the teacher model based on Eq. 1.

3.2. Asymmetric Teacher

This section elaborates on the proposed asymmetric teacher, dubbed **A-Teacher**, designed for refining the pseudo labels generated by the vanilla single-frame detector. As illustrated in Fig. 2, A-Teacher consists of three pivotal components: 1) the **P**ropagation-based **B**ox **A**ggregation (**PBA**) module focuses on improving the quality of the detected boxes in the current frame by propagating the boxes to supporting frames and then extracting target-related information. 2) the **D**reaming-based **B**ox **A**ggregation (**DBA**) module is introduced to address the false negatives in the current frame by aggregating detected boxes in supporting frames as dreaming queries and then verifying their presence in the current timestamp. The word "dreaming" is used to convey the idea that DBA creates ex nihilo for the current frame. 3) the **S**patio-**T**emporal feature **A**ggregation (**STA**) module is proposed to alleviate the objects neglected in both the current frame and any supporting frames by fusing point cloud features from all frames using deformable attention.

3.2.1 Propagation-based Box Aggregation

For objects detected in the current frame, we endeavor to improve its quality, *e.g.,* heading and size, by fusing its representations from both past ($t - k \sim t - 1$) and future ($t + 1 \sim t + k$) supporting frames. Prior to merging the features of a specific object, the prerequisite is to find its locations in other frames. An intuitive but cost choice is to use a tracker to assign unique identities to each object. However, in pursuit of efficiency, we propose realizing this function by spatial-aware cross-attention.

In specific, given the predicted 3D boxes of each frame, we first extract the corresponding appearance features using Voxel RoI pooling [7],

$$\mathbf{f}_{\mathrm{i}}^{\mathrm{j}} = \mathrm{RoIPooling}(\mathbf{b}_{\mathrm{i}}^{\mathrm{j}}, \mathbf{F}^{\mathrm{j}}) \in \mathbb{R}^{\mathrm{h}}, \tag{2}$$

where $j \in (t-k, t+k)$ indicates a timestamp. $\mathbf{b}_{\mathrm{i}}^{\mathrm{j}}$ and sparse voxel features $\mathbf{F}^{\mathrm{j}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{spa}}^{\mathrm{j}} \times \mathrm{d}_{\mathrm{spa}}}$ denote the i_{th} detected box and voxel feature of the j_{th} frame, respectively. To incorporate more geometry and semantic information, we extend the appearance embedding $\mathbf{f}_{\mathrm{i}}^{\mathrm{j}}$ with the predicted classification score $\mathrm{s}_{\mathrm{i}}^{\mathrm{j}}$, object category $\mathrm{c}_{\mathrm{i}}^{\mathrm{j}}$ and 3D size $\mathbf{b}_{\mathrm{i}}^{\mathrm{j}}$ of the box,

$$\hat{\mathbf{f}}_{\mathrm{i}}^{\mathrm{j}} = [\mathbf{f}_{\mathrm{i}}^{\mathrm{j}}, \mathrm{Linear}(\mathrm{s}_{\mathrm{i}}^{\mathrm{j}}), \mathrm{Linear}(\mathrm{c}_{\mathrm{i}}^{\mathrm{j}}), \mathrm{Linear}(\mathbf{b}_{\mathrm{i}}^{\mathrm{j}})] \in \mathbb{R}^{\mathrm{d}}, \tag{3}$$

where Linear is a linear projection layer. $[\cdot, \cdot]$ indicates concatenating the embeddings along the channel dimension. Then, for a box $\mathbf{b}_{\mathrm{i}}^{\mathrm{t}}$ in the current frame t, we aggregate its temporal features by spatial-aware cross-attention,

$$\bar{\mathbf{f}}_{\mathrm{i}}^{\mathrm{t}} = \mathrm{Softmax}\left(\frac{\hat{\mathbf{f}}_{\mathrm{i}}^{\mathrm{t}}\mathbf{K}^{\mathrm{j}\neq\mathrm{t}}}{\sqrt{d}} - \tau_{\mathrm{i}}^{\mathrm{t}}\mathbf{D}_{\mathrm{i}}^{\mathrm{t}}\right)\mathbf{V}^{\mathrm{j}\neq\mathrm{t}}, \tag{4}$$

where $\mathbf{K}^{\mathrm{j}\neq\mathrm{t}}, \mathbf{V}^{\mathrm{j}\neq\mathrm{t}} \in \mathbb{R}^{\mathrm{N}^{\mathrm{j}\neq\mathrm{t}} \times \mathrm{d}}$ and coefficient $\tau_{\mathrm{i}}^{\mathrm{t}}$ are generated by mapping the box embeddings (Eq. 3) with a linear layer on $\hat{\mathbf{f}}_{\mathrm{i}}^{\mathrm{j}\neq\mathrm{t}}$ and $\hat{\mathbf{f}}_{\mathrm{i}}^{\mathrm{j}=\mathrm{t}}$ respectively. $\mathrm{Softmax}$ denotes softmax layer. $\mathbf{D}_{\mathrm{i}}^{\mathrm{t}}$ is the distances between $\mathbf{b}_{\mathrm{i}}^{\mathrm{t}}$ and the boxes correspond with $\mathbf{K}^{\mathrm{j}\neq\mathrm{t}}$ and $\mathbf{V}^{\mathrm{j}\neq\mathrm{t}}$, which is introduced to inject the spatial relation to attention learning. Finally, we pass the concatenated features $[\hat{\mathbf{f}}_{\mathrm{i}}^{\mathrm{t}}, \bar{\mathbf{f}}_{\mathrm{i}}^{\mathrm{t}}]$ into linear layers to obtain the refined classification score $\hat{\mathrm{s}}_{\mathrm{i}}^{\mathrm{t}}$ and the offset $\mathbf{\Delta}_{\mathrm{i}}^{\mathrm{t}}$ of the original box $\mathbf{b}_{\mathrm{i}}^{\mathrm{t}}$. Eventually, the box is updated to $\mathcal{P}_{\mathrm{pba}}$ by the predicted offset. Please refer to [13] for more details about merging $\mathbf{b}_{\mathrm{i}}^{\mathrm{t}}$ and $\mathbf{\Delta}_{\mathrm{i}}^{\mathrm{t}}$.

3.2.2 Dreaming-based Box Aggregation

To address objects that were overlooked in the current timestamp but successfully detected in supporting past or future frames, we introduce the Dreaming-based Box Aggregation module to mitigate these false negatives. The essence is propagating the *unmatched boxes* from the supporting frames to retrieve and confirm their presence in the current frame. As previously mentioned, one intuitive way to achieve this goal is using a tracker for box propagation and association. However, due to its computational cost, this approach was ruled out as an option. Similar to the propagation-based box aggregation described in Sec. 3.2.1, it can also be accomplished through a straightforward cross-attention between $\hat{\mathbf{f}}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}} \in \mathbb{R}^{\mathrm{M}\times\mathrm{d}}$ and $\mathbf{F}^{\mathrm{t}}$ (resembling a reverse process of Sec. 3.2.1). Nevertheless, it's important to note that an object may be detected in different supporting frames, the mentioned cross-attention between $\hat{\mathbf{f}}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}$ and $\mathbf{F}^{\mathrm{t}}$ would introduce redundant computation.

For the sake of efficiency and to encode richer texture and motion information, we first squeeze the *unmatched boxes* in supporting frames to a predefined dreaming queries $\mathbf{Q}_{\mathrm{dre}}^{\mathrm{t}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{dre}}\times\mathrm{d}}$, where $N_{\mathrm{dre}} \ll M$. Specifically, we first conduct cross-attention between $\mathbf{Q}_{\mathrm{dre}}^{\mathrm{t}}$ and $\hat{\mathbf{f}}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}$,

$$\bar{\mathbf{Q}}_{\mathrm{dre}}^{\mathrm{t}} = \mathrm{Softmax}(\frac{\mathbf{Q}_{\mathrm{dre}}^{\mathrm{t}}\mathbf{K}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}}{\sqrt{d}})\mathbf{V}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}, \tag{5}$$

where $\mathbf{K}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}, \mathbf{V}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}} \in \mathbb{R}^{\mathrm{M}\times\mathrm{d}}$ are obtained by applying a Linear layer on $\hat{\mathbf{f}}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}$. The next step is propagating the dreaming queries to the current frame to recall the undetected boxes. In specific, we project unmatched boxes $\mathcal{B}_{\mathrm{u}}^{\mathrm{j}\neq\mathrm{t}}$ to the current timestamp based on transformation matrix

$\mathbf{E}_{\mathrm{t+k\to t}}$. Taking the $t+k$ frame as an example,

$$\mathcal{B}_{\mathrm{u}}^{\mathrm{t}} = \mathbf{E}_{\mathrm{t+k\to t}}\mathcal{B}_{\mathrm{u}}^{\mathrm{t+k}}, \tag{6}$$

where $\mathbf{E}_{\mathrm{t+k\to t}}$ is the transformation matrix that maps the coordinates between different frames. Given that an instance with the same identity in adjacent supporting frames may have close spatial locations, their projected boxes in the current timestamp tend to overlap. We thus employ Non-Maximum suppression (NMS [31]) on $\mathcal{B}_{\mathrm{u}}^{\mathrm{t}}$ to generate the sparse pseudo counterpart $\hat{\mathcal{B}}_{\mathrm{u}}^{\mathrm{t}}$ for the current timestamp. Subsequently, Voxel RoI pooling and linear layers are applied to $\mathbf{F}^{\mathrm{t}}$ based on $\hat{\mathcal{B}}_{\mathrm{u}}^{\mathrm{t}}$ to obtain the key $\mathbf{K}_{\mathrm{pse}}^{\mathrm{t}}$ and value $\mathbf{V}_{\mathrm{pse}}^{\mathrm{t}}$ associated with the pseudo boxes. We enlarge the boxes to encode more context information. Finally, a DETR-like decoder is used to predict the new boxes,

$$\mathcal{P}_{\mathrm{dba}} = \mathrm{Decoder}(\bar{\mathbf{Q}}_{\mathrm{dre}}^{\mathrm{t}}, \mathbf{K}_{\mathrm{pse}}^{\mathrm{t}}, \mathbf{V}_{\mathrm{pse}}^{\mathrm{t}}). \tag{7}$$

3.2.3 Spatio-Temporal Deformable Aggregation

When dealing with objects located far away from the ego vehicle or those that are partially occluded, detection based on a single frame may not be able to obtain sufficient information. In such seniors, the objects might be omitted in both current and supporting frames. To address this challenge, we propose to align the features of multiple frames to construct a more comprehensive representation of the objects. To compensate for the motion information, we align the BEV feature of different temporal frames from the vanilla teacher to the current timestamp with the transformation matrix $\mathbf{E}_{\mathrm{t'\to t}}$,

$$\mathbf{F}_{\mathrm{bev}}^{\mathrm{t'\to t}} = \mathbf{E}_{\mathrm{t'\to t}}\mathbf{F}_{\mathrm{bev}}^{\mathrm{t'}} \in \mathbb{R}^{(\mathrm{L}*\mathrm{d_{spa}})\times\mathrm{W}\times\mathrm{H}}. \tag{8}$$

For methods utilizing sparse-conv [47] like our baseline PV-RCNN [35], we only save the sparse features of the supporting frames to save memory. Before the temporal transformation, we first convert the sparse features to BEV representation by padding empty voxels and squeezing the feature along the height dimension. The transformation matrix is obtained based on the ego vehicle's poses. While this approach is effective for static objects, it is less suitable for moving objects due to their relative motions. Therefore, we first predict the offset between the supporting and the current timestamp,

$$\mathbf{O} = \mathrm{Conv}([\mathbf{F}_{\mathrm{bev}}^{\mathrm{t-k\to t}}, ..., \mathbf{F}_{\mathrm{bev}}^{\mathrm{t}}, ..., \mathbf{F}_{\mathrm{bev}}^{\mathrm{t+k\to t}}]). \tag{9}$$

Then we aggregate the BEV features of different frames via deformable convolution,

$$\hat{\mathbf{F}}_{\mathrm{bev}}^{\mathrm{t}} = \mathrm{DeformConv}([\mathbf{F}_{\mathrm{bev}}^{\mathrm{t-k\to t}}, ..., \mathbf{F}_{\mathrm{bev}}^{\mathrm{t}}, ..., \mathbf{F}_{\mathrm{bev}}^{\mathrm{t+k\to t}}], \mathbf{O}). \tag{10}$$

Finally, we apply a center-based detection head [54] to obtain the final detection results $\mathcal{P}_{\mathrm{sta}}$.

3.3. Pseudo Label Generation

Given the pseudo boxes generated by our proposed attention-based refinement modules, the subsequent phase involves merging and filtering these boxes to retain those of high quality. First, we design the priority-guided NMS to eliminate redundant predictions. Specifically, when exploiting non-maximum suppression, the priority parameter α_{i} is added to the confidence scores of the predicted boxes from each refinement module. According to our design logic, the priority parameters are assigned values of 2, 1, and 0 for PBA, DBA, and STA, respectively. Then inspired by HSSDA [17], we introduce a dual-threshold strategy for each category to selectively choose high-quality boxes for copy-paste augmentation. In particular, for each category, we apply thresholds λ^{high}, λ^{low} on the classification scores of each candidate box. Objects with a score above λ^{high}, are considered more informative and are employed in copy-paste augmentation, which is crucial for LiDAR-based 3D detection. Conversely, objects with classification scores falling below λ^{low} are excluded to guarantee the quality of pseudo labels. All boxes with higher classification scores than λ^{low} are retained for semi-supervised training.

3.4. Loss

Loss for the Vanilla Teacher. For the supervised training on labeled data of the vanilla teacher, the loss function remains consistent with our baseline PV-RCNN [35],

$$\mathcal{L}^{\mathrm{l}} = \mathcal{L}_{\mathrm{rpn}}(\tilde{\mathbf{y}}, \mathbf{y}^{\mathrm{l}}) + \mathcal{L}_{\mathrm{rcnn}}(\tilde{\mathbf{y}}, \mathbf{y}^{\mathrm{l}}), \tag{11}$$

where $\tilde{\mathbf{y}}$ is the prediction and $\mathbf{y}^{\mathrm{l}}$ is the ground truth of labeled data. $\mathcal{L}_{\mathrm{rpn}}$ and $\mathcal{L}_{\mathrm{rcnn}}$ indicates the losses used in PV-RCNN [35]. Please refer to PV-RCNN [35] for details.

Loss for the Refinement Model. For the training of the proposed refinement model on the labeled data, its loss contains three components corresponding to each module,

$$\mathcal{L}^{\mathrm{r}} = \mathcal{L}_{\mathrm{pba}} + \mathcal{L}_{\mathrm{dba}} + \mathcal{L}_{\mathrm{sta}}, \tag{12}$$

where $\mathcal{L}_{\mathrm{pba}}$, $\mathcal{L}_{\mathrm{dba}}$, $\mathcal{L}_{\mathrm{sta}}$ denote losses for the propagation-based aggregation, dreaming-based aggregation and spatio-temporal aggregation modules, respectively. $\mathcal{L}_{\mathrm{pba}}$ contains the offset loss $\mathcal{L}_{\mathrm{pba}}^{\mathrm{o}}$ and confidence loss $\mathcal{L}_{\mathrm{pba}}^{\mathrm{c}}$,

$$\begin{aligned}\mathcal{L}_{\mathrm{pba}}^{\mathrm{o}} &= \mathrm{SmoothL}_1(\boldsymbol{\Delta}, \boldsymbol{\Delta}_{\mathrm{gt}}),\\ \mathcal{L}_{\mathrm{pba}}^{\mathrm{c}} &= \mathrm{BinaryCrossEntropy}(\mathbf{s}, \mathbf{s}_{\mathrm{gt}}),\end{aligned} \tag{13}$$

where $\boldsymbol{\Delta}_{\mathrm{gt}}$ and $\mathbf{s}_{\mathrm{gt}}$ denote the corresponding label. Please refer to [13] for more details. We obtain $\mathcal{L}_{\mathrm{pba}}$ by summing the offset and confidence losses $\mathcal{L}_{\mathrm{pba}} = \mathcal{L}_{\mathrm{pba}}^{\mathrm{o}} + \mathcal{L}_{\mathrm{pba}}^{\mathrm{c}}$.

For $\mathcal{L}_{\mathrm{dba}}$, we utilize the Hungarian algorithm [12] for object assignment, subsequently followed by the loss computation as outlined in [42]. As in [54], $\mathcal{L}_{\mathrm{sta}}$ contains both heatmap loss and regression loss.

Method	Sensors	Veh. (L1)		Veh. (L2)		Ped. (L1)		Ped. (L2)		Cyc. (L1)		Cyc. (L2)		L1
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP
PV-RCNN[35]	L	**48.5**	**46.2**	**45.5**	**43.3**	**30.1**	**15.7**	**27.3**	**15.9**	**4.5**	**3.0**	**4.3**	**2.9**	**27.7**
DetMatch[24]	LC	52.2	51.1	48.1	47.2	39.5	18.9	35.8	17.1	-	-	-	-	-
HSSDA†[17]	L	56.4	53.8	**49.7**	47.3	40.1	20.9	33.5	17.5	29.1	20.9	27.9	20.0	41.9
A-Teacher (ours)	L	**56.5**	**54.5**	49.2	**47.5**	**48.1**	**27.3**	**40.8**	**23.1**	**35.1**	**27.1**	**33.7**	**26.1**	**46.6**
Improvements	-	+8.0	+8.3	+3.7	+4.2	+18.0	+11.6	+13.5	+7.2	+30.6	+24.1	+29.4	+23.2	+18.9

Table 1. **Comparision with state-of-the-art methods on Waymo.** We use † to denote the test-time-augmentation (TTA).

Loss for the Semi-supervised Training. For training on the unlabeled data, the student model exploits pseudo labels $\mathbf{y}^{\mathrm{u}}$ generated by the teacher model for calculating the loss,

$$\mathcal{L}^{\mathrm{u}} = \sum_{\mathrm{j}} \omega_{\mathrm{j}} \mathcal{L}_{\mathrm{rpn}}(\tilde{\mathbf{y}}^{\mathrm{u}}_{\mathrm{j}}, \mathbf{y}^{\mathrm{u}}_{\mathrm{j}}) + \omega_{\mathrm{j}} \mathcal{L}_{\mathrm{rcnn}}(\tilde{\mathbf{y}}^{\mathrm{u}}_{\mathrm{j}}, \mathbf{y}^{\mathrm{u}}_{\mathrm{j}}), \qquad (14)$$

where ω_{j} denotes the weight coefficient. In our method, we consider the classification score as the weight coefficient to measure the quality of pseudo labels.

4. Experiment

4.1. Experimental Setup

We train and evaluate our method on the large-scale Waymo [38] dataset, which contains a total of 1150 scenes with 798 ones for training, and the other 202 ones for validation. For fair comparisons, we use the same labeled and unlabeled splits as HSSDA [17] in training. Specifically, 7 sequences(1%) (1388 point cloud scenes) of the 798 training sequences are selected as labeled data. We evaluate our method with the official metrics provided by Waymo [38] including average precision (AP) and average precision weighted by heading (APH) in LEVEL 1 (L1) and LEVEL 2 (L2) for *Vehicle*, *Pedestrian* and *Cyclist*. For supervised learning, we follow the official setting of PV-RCNN[35] to train the vanilla teacher on the 1% labeled data. For the efficiency of training, we only use the previous and last timestamp (*i.e.,* $k = 1$) to construct supporting frames. At the training stage, we use Adam with a one-cycle learning rate schedule. Specifically, we train the teacher and refinement model for 30 epochs in labeled data and 10 epochs for semi-supervised learning. The learning rate is 0.01. To enhance the robustness of our refinement model, we conduct data augmentations including random applying jittering noise to candidate boxes and random dropping point clouds within candidate boxes.

4.2. Comparison with State-of-the-arts

We compare the proposed A-Teacher with recent state-of-the-art (SOTA) semi-supervised approaches that also use PV-RCNN [35] as the baseline and evaluate on Waymo [38] dataset. As shown in Tab. 1, our semi-supervised framework obtains gains of 8.0 and 18.9 on $\mathrm{Veh.(L1)\ mAP}$ and overall mAP over the supervised baseline PV-RCNN [35]. Our method also surpasses previous SOTA methods DetMatch [24] and HSSDA [17] on almost all metrics. Notably, DetMatch [24] is multimodal, leveraging camera and LiDAR data, while HSSDA [17] employs test time augmentation (TTA) with random flip, rotate, and scaling for pseudo label generation. Differently, we neither use visual information from images nor exploit complex TTA. Even with such a simpler setting than DetMatch [24] and HSSDA [17], our model still shows superior performances, demonstrating the effectiveness of our framework. Notably, compared with HSSDA which exploits test time augmentation (TTA), the proposed method saves 15.9 % training resources, *i.e.,* 71.1 V100 GPU hours of our A-Teacher *v.s.* 84.6 V100 GPU hours of HSSDA [17]. Specifically, the computational requirement of our refinement module amounts to $\mathbf{0.37}\times$ that of the teacher model. In comparison, HSSDA [17] employs test-time augmentation for refinement, incurring a computational expense that is $\mathbf{2}\times$ of the teacher model. In the following sections, we conduct more experiments to reveal the significance of each proposed component. The ablation study is conducted on "1% labeled + 4% unlabeled" setting unless otherwise specified.

4.3. Component-wise Analysis

To demonstrate the superiority of our online asymmetric architecture, we intentionally degenerate our model to its online symmetric and offline asymmetric counterparts. As depicted in Tab. 2 ②, we remove the refinement model to construct the online symmetric counterpart. The results reveal that without temporal information, the online symmetric counterpart degrades the performances of all categories (② *v.s.* ⑩). The offline asymmetric counterpart, which refrains from updating the teacher model, unsurprisingly yields inferior results compared to our proposed framework (③ *v.s.* ⑩)), primarily due to the limited capability of the teacher model. Then, we conduct a detailed analysis of each component within the proposed refinement model. In comparison to the baseline (①), the performance notably enhances as components are incorporated (④-⑩). Furthermore, PBA (④), DBA (⑤), and STA (⑥) contribute gains of 5.5, 4.5, and 4.9 on $\mathrm{Veh.(L1)mAP}$, respectively, further affirming the effectiveness of our method.

#	PBA	DBA	STA	Up	Veh. (L1)		Veh. (L2)		Ped. (L1)		Ped. (L2)		Cyc. (L1)		Cyc. (L2)	
					mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
①	-	-	-	-	48.5	46.2	45.5	43.3	30.1	15.7	27.3	15.9	4.5	3.0	4.3	2.9
②	-	-	-	On	52.4	50.2	45.5	43.5	36.6	23.7	30.8	20.0	10.5	7.9	10.1	7.6
③	✓	✓	✓	Off	52.9	50.6	45.9	44.1	38.2	24.1	32.1	20.2	8.5	6.8	8.2	6.5
④	✓	-	-	On	54.0	51.7	46.9	44.9	40.0	24.9	33.7	20.1	11.8	9.0	11.4	8.7
⑤	-	✓	-	On	53.0	50.8	45.9	44.0	41.7	25.8	35.1	21.7	11.7	9.1	11.3	8.7
⑥	-	-	✓	On	53.4	51.1	46.4	44.3	41.3	24.9	34.8	21.0	13.5	10.6	13.0	10.2
⑦	✓	✓	-	On	53.5	51.2	46.5	44.5	42.9	27.1	36.2	22.9	13.9	9.9	13.3	9.6
⑧	✓	-	✓	On	54.1	51.8	46.9	45.0	42.7	27.0	36.1	22.7	**15.0**	**11.2**	**14.4**	**10.8**
⑩	✓	✓	✓	On	**54.4**	**52.1**	**47.3**	**45.3**	**43.1**	**27.4**	**36.4**	**23.1**	14.7	11.1	14.2	10.7

Table 2. **Component-wise Analysis.** ① is the results of training with labeled data. ② and ③ degenerate our framework to online symmetric and offline asymmetric counterparts. The other experiments show the influence of each proposed component. "Up" denotes the EMA.

Training Data	Veh. (L1)		Veh. (L2)		Ped. (L1)		Ped. (L2)		Cyc. (L1)		Cyc. (L2)		L1
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP
Baseline (1%)	48.5	46.2	45.5	43.3	30.1	15.7	27.3	15.9	4.5	3.0	4.3	2.9	27.7
5% (1+4%)	54.4	52.1	47.3	45.3	43.1	27.4	36.4	23.1	14.7	11.1	14.2	10.7	37.4
Improvement	+5.9	+5.9	+1.8	+2.0	+13.0	+11.7	+9.1	+7.2	+10.2	+8.1	+9.9	+7.8	+9.7
20% (1+19%)	55.8	53.6	48.6	46.7	45.5	31.7	38.5	26.8	27.7	22.1	26.6	21.3	43.0
Improvement	+7.3	+7.4	+3.1	+3.4	+15.4	+16.0	+11.2	+10.9	+23.2	+19.1	+22.3	+18.4	+15.3
100% (1+99%)	56.5	54.5	49.2	47.5	48.1	27.3	40.8	23.1	35.1	27.1	33.7	26.1	46.6
Improvement	+8.0	+8.3	+3.7	+4.2	+18.0	+11.6	+13.5	+7.2	+30.6	+24.1	+29.4	+23.2	+18.9

Table 3. **Influence of unlabeled data volume.** Keeping the labeled data unchanged, we increase the unlabeled data to see the impact.

Model	Veh. (L1)		Ped. (L1)	
	mAP	mAPH	mAP	mAPH
Second [47] (1%)	39.8	38.7	22.2	11.1
5%(1% +4%)	43.6	42.7	28.9	13.8
Improvement	+3.8	+4.0	+6.7	+2.7
Voxel-RCNN [7] (1%)	50.6	49.5	43.6	32.0
5%(1% +4%)	55.4	54.4	49.4	35.7
Improvement	+4.8	+4.9	+5.8	+3.7

Table 4. **Apply our framework to other detection methods.**

4.4. Further Analysis

Different Volume of Unlabeled Data. We then conduct experiments to study the impact of varying volumes of unlabeled data. As depicted in Tab. 3, when training on 1% labeled data, the baseline achieves 48.5 $\mathrm{Veh.(L1)}$ mAP and 27.7 overall $\mathrm{mAP(L1)}$, respectively. When providing 4% unlabeled data to the model, our A-Teacher achieves gains of 5.9 and 9.7 on $\mathrm{Veh.(L1)}$ mAP and overall $\mathrm{mAP(L1)}$. More unlabeled data consistently improves our model, particularly benefiting the challenging categories of *Pedestrian* and *Cyclist*. Experimental results affirm the effectiveness of our approach in assimilating knowledge from unlabeled data, where even a small amount of 4% unlabeled data brings considerable gains. It's worth noting that the observed enhancement rate is more pronounced for challenging categories like *Pedestrian* and *Cyclist* compared with *Vehicle*, potentially stemming from an imbalanced distribution of categories. We leave this to future study.

Apply A-Teacher to other Methods. To verify the generalization of the proposed A-Teacher, we apply our framework to other representative detection methods including Voxel-RCNN [7] and Second [47]. Notably, the baseline PV-RCNN [35] in previous experiments are point-voxel hybrid architecture. Differently, both of Voxel-RCNN [7] and Second [47] adopt the voxel-based design. For the sake of simplicity, we exclusively conduct experiments with "1% labeled + 4% unlabeled" data setting. As illustrated in Tab. 4, with our proposed semi-supervised framework, both of Second and Voxel-RCNN achieve impressive improvements, which proves the generalization of our method.

Quality of Pseudo Labels. As discussed earlier and qualitative analysis in Fig. 3, PBA enhances the quality of candidate boxes (Fig. 3 left) detected by the single-frame detector in the current frame. DBA and STA successfully recall ignored boxes of the current frame (Fig. 3 mid). Additionally, our refinement model effectively eliminates false positives (Fig. 3 right). Furthermore, it is observed that most semi-supervised 3D object detection approaches typically employ a pre-defined threshold on classification scores to filter pseudo labels. Hence, we conduct experiments to analyze the quality of pseudo labels under different filter thresholds. As depicted in Fig. 4, it is evident that our A-Teacher exhibits superior precision compared with the vanilla teacher across different thresholds. This observation underscores the efficacy and robustness of our refinement model. The analyses presented in Fig. 4 and Fig. 3 affirm the effectiveness of A-Teacher in enhancing pseudo labels.

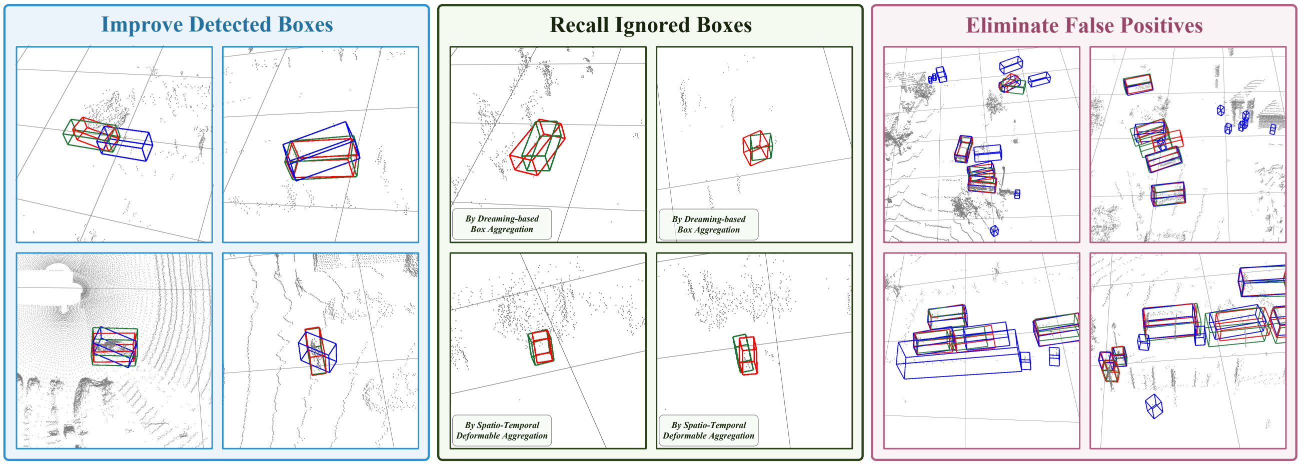

Improve Detected Boxes
Recall Ignored Boxes
Eliminate False Positives
By Dreaming-based Box Aggregation
By Dreaming-based Box Aggregation
By Spatio-Temporal Deformable Aggregation
By Spatio-Temporal Deformable Aggregation

Figure 3. **Pseudo labels visualization.** The **green** boxes represent the ground truth, the **blue** boxes denote the candidate predictions generated by the vanilla teacher, and the **red** boxes indicate the pseudo labels generated by the proposed attention-based refinement model.

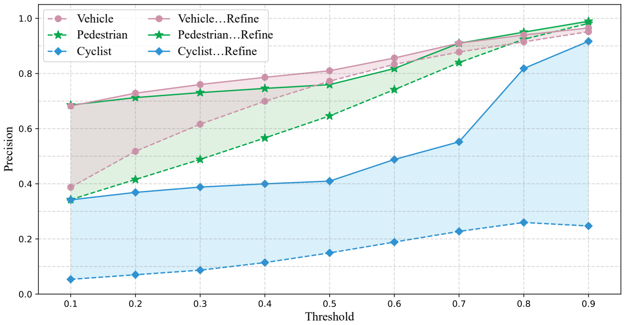


Figure 4. **Precision of pseudo labels.** We compare the quality of pseudo labels generated by the vanilla teacher (dashed lines) and that after the refinement model (solid lines) under different filter thresholds applied to classification scores. The shadow region indicates the improvements brought by our refinement model.

Model	Veh. (L1)		Ped. (L1)		Cyc. (L1)	
	mAP	mAPH	mAP	mAPH	mAP	mAPH
PV.(4f)	49.8	47.3	37.5	18.3	9.6	5.8
5%(1%+4%)	55.8	53.4	41.9	20.9	20.0	19.3
Improvement	+6.0	+6.1	+4.4	+2.6	+10.4	+13.5

Table 5. **Apply our framework to the multi-frame method.** We extend the student, *i.e.,* single frame PV-RCNN (PV.) [35], to its 4 frames version and keep the refinement model unchanged.

Extension to Multi-frame 3D Object Detection. Following the common settings in the LiDAR-based semi-supervised detection methods, we use a single-frame object detector as the student model in previous experiments and analyses. It is also known that multi-frame input is indispensable to achieve state-of-the-art performance. Therefore, we apply the proposed A-Teacher to the PV-RCNN of the 4-frame version to demonstrate the effectiveness of our framework. As shown in Tab. 5, even with the multi-frame student model, our A-Teacher can still bring clear promotion, which implies the universality of our approach and the potential to apply to most advanced 3D detection methods.

Methods	Veh. (AP)	Ped. (AP)	Cyc. (AP)	mAP
Baseline (labeled only)	71.19	26.44	58.04	51.89
NoiseDet [6]	75.26	37.96	60.77	58.00
A-Teacher (ours)	**77.42**	**39.49**	**64.65**	**60.52** (**+8.63**)

Table 6. Results on ONCE dataset ("Small" setting).

Extension to ONCE dataset. To further demonstrate our method's generalization, we conduct experiments on the ONCE dataset [21]. As shown in Tab. 6, our method achieves a mAP gain of 8.63 overall, reaffirming its efficacy. Notably, it surpasses the most recent paper NoiseDet [6].

5. Conclusion

In this paper, we propose the first online asymmetric semi-supervised framework for LiDAR-based 3D object detection. The cores of our success are the efficient refinement model and the capability of accessing future frames with the help of the masterly designed attention-based refinement model. Analytical experiments exhibit the precise pseudo labels generated by A-Teacher and the reason behind them. Furthermore, extension experiments demonstrate the effectiveness and generality of our A-Teacher in improving different detectors with single- or multi-frame input.

Acknowledgements. This work was supported in part by the National Key Research and Development Program of China (Grant No. 2022YFC3321003), the Beijing Natural Science Foundation (Grant No. JQ22014, L223003), the Natural Science Foundation of China (Grant No. U22B2056, 62192782, 62036011, U2033210, 62102417, 62202470, 62302501), the Project of Beijing Science and technology Committee (Project No. Z231100005923046). Jin Gao was also supported in part by the Youth Innovation Promotion Association, CAS.

References

[1] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 3

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 1

[3] Changrui Chen, Kurt Debattista, and Jungong Han. Semi-supervised object detection via virtual category learning. *arXiv preprint arXiv:2207.03433*, 2022. 2

[4] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *ECCV*, pages 680–697. Springer, 2022. 2

[5] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, pages 21674–21683, 2023. 1

[6] Zehui Chen, Zhenyu Li, Shuo Wang, Dengpan Fu, and Feng Zhao. Learning from noisy data for semi-supervised 3d object detection. In *ICCV*, pages 6929–6939, 2023. 3, 8

[7] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*, pages 1201–1209, 2021. 2, 4, 7

[8] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *CVPR*, pages 8458–8468, 2022. 2

[9] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection. *arXiv preprint arXiv:2304.12315*, 2023. 2

[10] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3d object detection, 2023. 1

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1

[12] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[13] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 1, 4, 5

[14] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. Rethinking pseudo labels for semi-supervised object detection. In *AAAI*, pages 1314–1322, 2022. 3

[15] Jinyu Li, Chenxu Luo, and Xiaodong Yang. Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. In *CVPR*, pages 17567–17576, 2023. 1

[16] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, et al. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *CVPR*, pages 17524–17534, 2023. 1

[17] Chuandong Liu, Chenqiang Gao, Fangcen Liu, Pengcheng Li, Deyu Meng, and Xinbo Gao. Hierarchical supervision and shuffle data augmentation for 3d semi-supervised object detection. In *CVPR*, pages 23819–23828, 2023. 1, 2, 3, 5, 6

[18] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021. 3

[19] Tao Ma, Xuemeng Yang, Hongbin Zhou, Xin Li, Botian Shi, Junjie Liu, Yuchen Yang, Zhizheng Liu, Liang He, Yu Qiao, et al. Detzero: Rethinking offboard 3d object detection with long-term sequential point clouds. *arXiv preprint arXiv:2306.06023*, 2023. 2

[20] Jiageng Mao, Minzhe Niu, Haoyue Bai, Xiaodan Liang, Hang Xu, and Chunjing Xu. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In *ICCV*, pages 2723–2732, 2021. 2

[21] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: Once dataset, 2021. 8

[22] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *ICCV*, pages 3164–3173, 2021. 2

[23] OpenAI. Gpt-4 technical report, 2023. 1

[24] Jinhyung Park, Chenfeng Xu, Yiyang Zhou, Masayoshi Tomizuka, and Wei Zhan. Detmatch: Two teachers are better than one for joint 2d and 3d semi-supervised object detection. In *ECCV*, pages 370–389. Springer, 2022. 2, 3, 6

[25] Yu Pei, Xian Zhao, Hao Li, Jingyuan Ma, Jingwei Zhang, and Shiliang Pu. Clusterformer: Cluster-based transformer for 3d object detection in point clouds. In *CVPR*, pages 6664–6673, 2023. 1

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 2

[28] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 2

[29] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *CVPR*, pages 6134–6144, 2021. 2

[30] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS*, 35:23192–23204, 2022. 1

[31] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971. 5

[32] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, pages 2743–2752, 2021. 2

[33] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: High-performance pillar-based 3d object detection. *arXiv preprint arXiv:2205.07403*, 2022. 1

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 2

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020. 2, 5, 6, 7, 8

[36] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *IJCV*, 131(2):531–551, 2023. 2

[37] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2, 3

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 1, 2, 6

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017. 2, 3

[40] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *CVPR*, pages 14615–14624, 2021. 2, 3

[41] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, pages 13520–13529, 2023. 2

[42] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 5

[43] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *CVPR*, pages 4568–4577, 2021. 3

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020. 3

[45] Jianyun Xu, Zhenwei Miao, Da Zhang, Hongyu Pan, Kaixuan Liu, Peihan Hao, Jun Zhu, Zhengyang Sun, Hongmin Li, and Xin Zhan. Int: Towards infinite-frames 3d detection with an efficient framework. In *ECCV*, pages 193–209. Springer, 2022. 1

[46] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, pages 3060–3069, 2021. 3

[47] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 5, 7

[48] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. Auto4d: Learning to label 4d objects from sequential point clouds. *arXiv preprint arXiv:2101.06586*, 2021. 2

[49] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, pages 1951–1960, 2019. 2

[50] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, pages 11040–11048, 2020. 1

[51] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *CVPR*, pages 1863–1872, 2021. 1

[52] Maosheng Ye, Shuangjie Xu, and Tongyi Cao. Hvnet: Hybrid voxel network for lidar based 3d object detection. In *CVPR*, pages 1631–1640, 2020. 2

[53] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Semi-supervised 3d object detection with proficient teachers. In *ECCV*, pages 727–743. Springer, 2022. 3

[54] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 2, 5

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16259–16268, 2021. 2

[56] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *CVPR*, pages 11079–11087, 2020. 3

[57] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *AAAI*, pages 3555–3562, 2021. 1

[58] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *CVPR*, pages 4081–4090, 2021. 3

[59] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2

[60] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *ECCV*, pages 496–513. Springer, 2022. 2