Semi-supervised 3D Object Detection with PatchTeacher and PillarMix

Xiaopei Wu[1, 2†], **Liang Peng**[1], **Liang Xie**[1], **Yuenan Hou**[2],
Binbin Lin[3*], **Xiaoshui Huang**[2*], **Haifeng Liu**[1], **Deng Cai**[1], **Wanli Ouyang**[2]

[1]State Key Lab of CAD&CG, Zhejiang University
[2]Shanghai AI Laboratory
[3]School of Software Technology, Zhejiang University
{wuxiaopei, pengliang}@zju.edu.cn

Abstract

Semi-supervised learning aims to leverage numerous unlabeled data to improve the model performance. Current semi-supervised 3D object detection methods typically use a teacher to generate pseudo labels for a student, and the quality of the pseudo labels is essential for the final performance. In this paper, we propose **PatchTeacher**, which focuses on partial scene 3D object detection to provide high-quality pseudo labels for the student. Specifically, we divide a complete scene into a series of patches and feed them to our PatchTeacher sequentially. PatchTeacher leverages the low memory consumption advantage of partial scene detection to process point clouds with a high-resolution voxelization, which can minimize the information loss of quantization and extract more fine-grained features. However, it is non-trivial to train a detector on fractions of the scene. Therefore, we introduce three key techniques, i.e., *Patch Normalizer*, *Quadrant Align*, and *Fovea Selection*, to improve the performance of PatchTeacher. Moreover, we devise **PillarMix**, a strong data augmentation strategy that mixes truncated pillars from different LiDAR scans to generate diverse training samples and thus help the model learn more general representation. Extensive experiments conducted on Waymo and ONCE datasets verify the effectiveness and superiority of our method and we achieve new state-of-the-art results, surpassing existing methods by a large margin. Codes are available at https://github.com/LittlePey/PTPM.

Introduction

Recent years have witnessed the rapid development of 3D object detection owing to the boom in deep learning. Many 3D detection methods (Shi et al. 2020; Yin, Zhou, and Krahenbuhl 2021; Wu et al. 2022; Chen et al. 2022; Liu et al. 2023b; Li et al. 2023) have achieved impressive performance on various leaderboards. The appealing performance of these 3D detectors heavily relies on the high-quality annotations of large-scale 3D datasets, which consume enormous human efforts and time. Semi-supervised learning (SSL), which leverages large quantities of low-cost and readily available unlabeled data, is gaining surging attention.

Current semi-supervised 3D detection methods usually follow the practice of 2D, leveraging an EMA teacher to generate better pseudo labels as the training process proceeds. However, the EMA approach requires the teacher to use the same setting as the student, which limits us from utilizing stronger teachers. Recently, some works have noticed this and tried to build a heterogeneous teacher based on or not based on the EMA teacher. For example, (Qi et al. 2021) use a multi-frame teacher to generate accurate pseudo labels for the single-frame student. (Yin et al. 2022) employs a TTA (test-time augmentation) teacher to supervise the single-forward student. (Peng et al. 2022) takes advantage of the LiDAR-based teacher to generate strong pseudo labels for the image-based student. However, the aforementioned methods only focus on complete scene detection. The huge amount of points of the complete scene limits their teachers from using high-resolution voxelization to generate more accurate pseudo labels.

To this end, we propose **PatchTeacher**, which leverages the low memory consumption advantage of partial scene detection to process point clouds with a high-resolution voxelization. It aims to break out the performance bottleneck of the detector trained on the complete scene and produce high-quality pseudo labels for the student. Concretely, we divide a complete scene into $N \times N$ patches and feed them to our PatchTeacher sequentially. Considering that the points of a patch are much fewer than a complete scene, we can use a very small voxel size for voxelization. The high-resolution voxelization enables the model to extract more fine-grained and discriminative features, which is of great benefit to high-performance 3D object detection.

However, it is non-trivial to train PatchTeacher on point cloud patches. *Firstly*, distant patches contain few positive samples. The total loss will be dominated by the negative classification loss of distant patches (see Equation 1 for more details and discussions). Therefore, we propose ***Patch Normalizer***. It normalizes the negative classification loss of different patches in the same frame with the same dominators to prevent the gradient of the classification branch from being dominated by distant patches. *Secondly*, to train all patches with the same detector, we need to shift them to the same coordinate system, ensuring they are in the same point cloud range. However, simply shifting patches from different quadrants to the same range will inevitably alter

*Corresponding author. † This work was done during his internship at Shanghai Artificial Intelligence Laboratory.

the radiation distribution of point clouds and thus increases the learning difficulty. To tackle this issue, ***Quadrant Align*** is introduced. It rotates patches from different quadrants to the same quadrant before shifting them to maintain the intrinsic characteristic of LiDAR point clouds. *Thirdly*, objects located at the edge of patches are difficult to be accurately predicted due to the truncation. We devise ***Fovea Selection***, which generates a series of border-overlapped but fovea-non-overlapped patches to address this edge truncated problem. Endowed with the above-mentioned strategies, the performance of PatchTeacher is highly boosted. As a model-agnostic design, our PatchTeacher can empower different students without modifying their architectures and improve semi-supervised learning.

Additionally, given pseudo labels produced by a teacher, how to help the model learn useful information from them is also essential. Weak-strong data augmentation is a promising strategy that has been proven in many previous literatures (Sohn et al. 2020a,b). It enforces the model to make consistent predictions for the weakly augmented and the strongly augmented unlabeled data and thus enables the model to learn useful information from the pseudo labels. Commonly, a stronger data augmentation can help the model learn more information from the pseudo annotations. Based on this analysis, we propose **PillarMix**, a strong data augmentation strategy that mixes pillars of different LiDAR scans. Specifically, we divide each LiDAR scan into $M \times M$ pillars and then cross-mix these pillars, ensuring adjacent pillars are always from different LiDAR scans. This approach produces strongly truncated point clouds on the edges of pillars, which can bring two merits. First, the blurred edge makes more hard samples which benefits the learning of the model on occluded or sparse samples. Second, it encourages the model to learn more general features under diverse surrounding environments, which plays a role in regularization. Experimental results reveal that our PillarMix can achieve impressive improvements in semi-supervised 3D object detection. To summarize, the contributions of this paper are listed as follows:

- We present ***PatchTeacher***, which explores the potential of partial scene detection with super high resolution to generate strong pseudo labels for semi-supervised 3D detection. In addition, three necessary techniques are devised to improve the performance of PatchTeacher.
- We propose ***PillarMix***, which cross-mixes pillars from different LiDAR scans to enrich the semi-supervised training data. It enforces the model to make consistent predictions given various incomplete point clouds and surrounding environments.
- Our overall pipeline significantly outperforms previous state-of-the-art methods. Extensive experiments on Waymo and ONCE datasets demonstrate the effectiveness and superiority of our method.

Related Work

Semi-Supervised Learning

Semi-Supervised Learning draws growing attention in a wide range of research areas. Since unlabeled data can be obtained more easily than labeled data, unlabeled data is far more than labeled data. Semi-supervised learning focuses on leveraging the labeled and unlabeled data to help the learning of the model. Current semi-supervised methods can be roughly divided into pseudo-label-based methods (Lee et al. 2013; Iscen et al. 2019; Arazo et al. 2020) and consistency regularization methods (Bachman, Alsharif, and Precup 2014; Sajjadi, Javanmardi, and Tasdizen 2016; Laine and Aila 2016). The former pre-trains a model on labeled data and uses the pre-trained model to infer the unlabeled data. A predefined threshold is used to filter out the high-quality pseudo labels, and these pseudo labels are used as the ground truth to re-train the model. The latter builds a regularization loss with unlabeled images, which encourages the model to generate similar predictions on different perturbations of the same image.

Semi-Supervised Object Detection

Semi-supervised object detection can also be divided into two groups: consistency-based methods (Jeong et al. 2019; Tang et al. 2021) and pseudo-labeling methods (Sohn et al. 2020b; Liu et al. 2021; Xu et al. 2021). For semi-supervised 3D object detection, SESS (Zhao, Chua, and Lee 2020) is the pioneering work based on consistency regularization. (Wang et al. 2021a) use the IoU prediction as a localization metric to filter poorly localized proposals. (Qi et al. 2021; Wang et al. 2021b) utilizes multi-frames to produce more accurate detection results. (Yin et al. 2022) enhances the teacher model to a proficient one with three careful designs. (Park et al. 2022) presents a flexible framework for joint semi-supervised learning on 2D and 3D modalities. (Leng et al. 2022) enriches semi-supervised training data with three pseudo-labeling data augmentation policies. (Liu et al. 2023a) devises a dynamic dual-threshold strategy and a shuffle data augmentation strategy to improve the performance of SSL. However, these works focus on complete scene detection, which can not leverage high-resolution voxelization to generate more accurate pseudo labels due to memory constrain. In this paper, we leverage partial scene detection to build a super high-resolution teacher for high-quality pseudo labels.

Methodology

Problem Definition

In semi-supervised 3D object detection, we are given a set of labeled data $\mathcal{L} = \{L_i\}_{i=1}^{N_l}$ and a set of unlabeled data $\mathcal{U} = \{U_i\}_{i=1}^{N_u}$, where N_l and N_u denote the amount of labeled and unlabeled frames, respectively. The objective of semi-supervised 3D object detection is to improve the performance of the model with both labeled and unlabeled data.

Framework Overview

As illustrated in Figure 1, our semi-supervised framework consists of two phases. In phase 1, we train a high-performance PatchTeacher. In phase 2, the student uses the pseudo labels produced by PatchTeacher for semi-supervised learning. The proposed PillarMix is used to further improve semi-supervised learning.

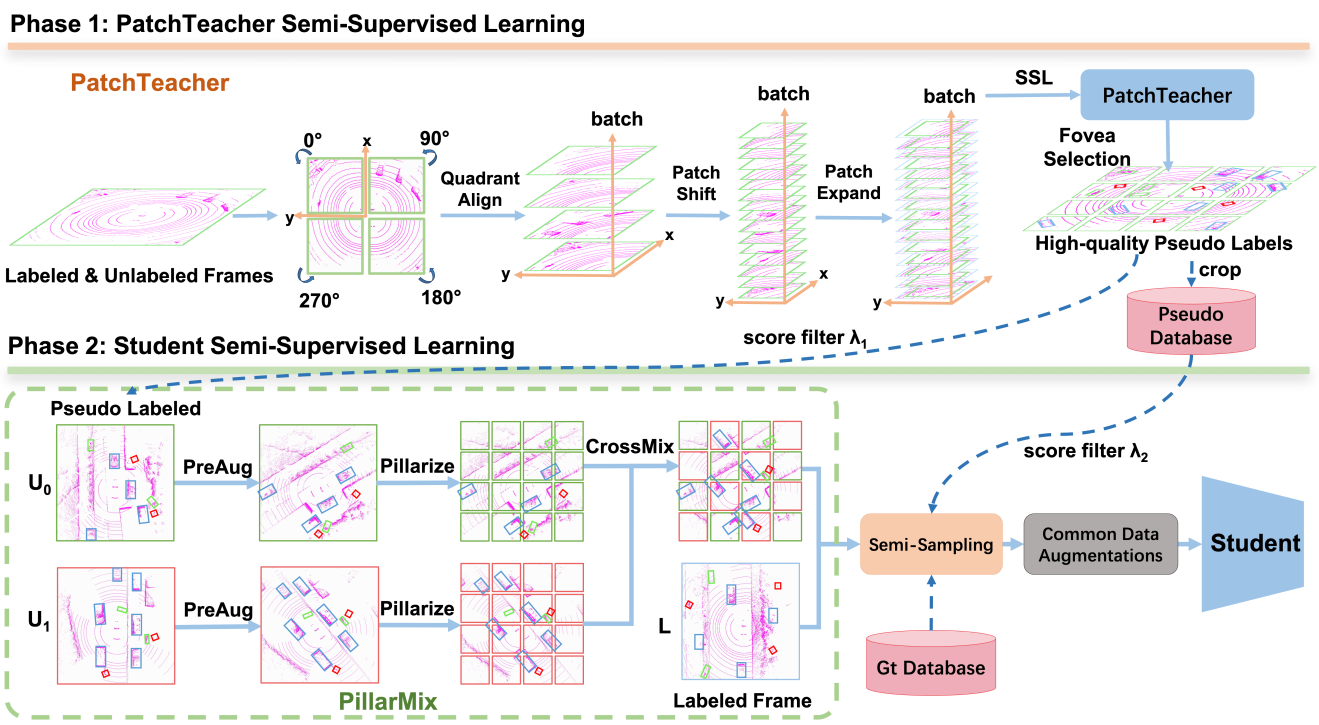

Figure 1: Our semi-supervised 3D object detection framework comprises two phases. In phase 1, we train a high-performance PatchTeacher. It focuses on partial scene detection, which enables a super high-resolution voxelization, achieving superior improvement. Three practical techniques and SSL are used to further boost the performance of our PatchTeacher. In phase 2, the high-quality pseudo labels produced by PatchTeacher is used to supervise the student model. Given pseudo labels, to make full use of them, we propose PillarMix, which mixes pillars of different LiDAR scans crossly, making a strong data augmentation. Then semi-sampling and common data augmentations are followed. Note that semi-sampling is the improved version we develop based on PseudoAugment. The details are provided in the implementation details.

PatchTeacher

Given a frame of point cloud $\mathcal{P}$, our PatchTeacher uses a divide-and-conquer manner to process it. Specifically, we partition $\mathcal{P}$ to $N \times N$ patches, which we denote $\{p_i\}_{i=1}^{N \times N}$, and then we process these patches sequentially. Each patch contains a small range of point clouds, which only consumes a low memory overhead. This enables us to use an extremely small voxel size to voxelize point clouds. PatchTeacher uses the same architecture as the student while focusing on the partial scene detection and leveraging high-resolution voxelization to achieve superior performance than the student. After processing all patches, we merge their predictions to generate the final predictions of $\mathcal{P}$. However, directly training PatchTeacher can only achieve limited gains. To this end, we propose three practical techniques to improve the performance of PatchTeacher.

Patch Normalizer. For 3D object detection, the RPN loss is usually formulated as follows:

$$\mathcal{L}_{\text{RPN}} = \frac{1}{N_{\text{fg}}} \sum_i \mathcal{L}_{\text{cls}}(p_i^a, c_i^*) + \frac{1}{N_{\text{fg}}} \mathbb{1}(c_i^* \geq 1) \sum_i \mathcal{L}_{\text{reg}}(\delta_i^a, t_i^*), \tag{1}$$

where N_{fg} is the number of foreground anchors, which serves as loss normalizer. p_i^a and δ_i^a are the outputs of classification and box regression branches, c_i^* and t_i^* are the classification label and regression targets respectively. $\mathbb{1}(c_i^* \geq 1)$ indicates regression loss in only calculated with foreground anchors. Due to the occlusion and sparsity properties of point clouds, foreground objects in distant patches are typically rare, resulting in a small normalizer (N_{fg}) and a huge amount of easy negatives. On the contrary, the foreground objects of nearby patches are usually crowded. Hence, the N_{fg} is large and there are lots of hard negatives that have a low overlap with the ground truth boxes. When using loss in Equation 1 to train PatchTeacher, gradients of the classification branch can be overwhelmed by easy negatives from distant patches whose loss normalizers are small. This may harm the training and result in poor performance.

To this end, we propose to normalize the classification losses of patches of the same LiDAR scan with the same normalizer, which is motivated by the complete scene detection framework. In complete scene detection, the loss of either distant or nearby negatives is normalized by total positive samples. In high-resolution partial scene detection, due to the memory constraint, we can not feed all patches in a frame to the network in an iteration, making it impossible to get the number of total positives. Considering the number of positive samples N_{fg} and ground-truth boxes are positively correlated, we can multiply the number of the total ground-truth boxes of a scene with a factor α to approximate the

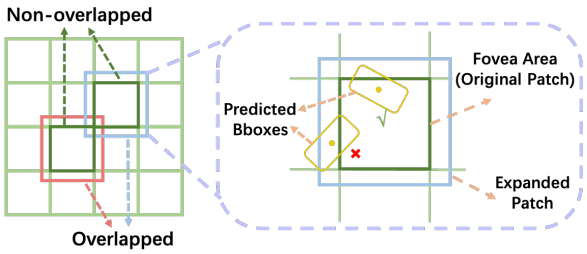

Figure 2: Illustration of Fovea Selection.

number of total positives. Formally, we define the Patch Normalizer as $N_{\text{p}} = \alpha * \sum_{j=1}^{N \times N} G_j$ ($\alpha = 3$ by default), where G_j is the num of ground truth boxes in the j^{th} patch. When training PatchTeacher, simply replacing the loss normalizer of classification loss from N_{fg} to N_{p} can obtain good results.

Quadrant Align. To train all patches with the same detector, we need to shift them to the same coordinate system, ensuring they are in the same point cloud range. However, simply shifting patches from different quadrants to the same range will inevitable alter the radiation distribution of point clouds which may increase the learning difficulty. To alleviate this issue, we align the patches from different quadrants to the same quadrant by anticlockwise rotating them around the origin point before shifting. In this way, the distribution consistency of point clouds in different patches can be guaranteed, making the training more effective. After *Quadrant Align*, we further divide the point clouds in the same quadrant into several patches evenly. To train them with the same model, we use the simple translation operation to align them to the same point cloud range. We call this *Patch Shift*.

Fovea Selection. Considering that the truncated point clouds caused by partition can increase the inference difficulty on the edges of the patches, we propose Fovea Selection strategy. Concretely, we expand the original non-overlapped patches by δ meters (δ = 2 by default), resulting in a series of overlapped patches, as shown in Figure 2. We define the original non-overlapped area of the expanded patch as its fovea area. When training PatchTeacher, all points and labels in expanded patches will be used. In the inference, the points in expanded patches are also used but we only select the predictions whose centers are in the fovea area of expanded patches to form the predictions of a complete scene. In this way, all predictions we use are based on the non-truncated point clouds of expanded patches. This method is especially useful for large objects, such as cars, and Table 5 provides the ablation study on Fovea Selection.

Semi-Supervised Learning on PatchTeacher. To generate stronger pseudo labels for the student of phase 2, we perform semi-supervised learning on the PatchTeacher in phase 1. The semi-supervised learning for PatchTeacher is similar to the student, and the pseudo labels are generated by PatchTeacher itself.

PillarMix

Given two LiDAR scans U_0 and U_1 and their pseudo labels P_0 and P_1. The output of our PillarMix is a fused LiDAR scan U_{mix} with its pseudo labels P_{mix}. Since the mixing of P_{mix} and U_{mix} is similar, here we only describe the mixing of U_{mix}. To increase the diversity of the fused LiDAR scan, we perform random rotation, random flipping and random scaling on U_0 and U_1 before pillarizing them. Then we divide U_0 and U_1 to $M \times M$ pillars evenly as shown in Figure 1. For pseudo label partition, a pseudo box belongs to a pillar when its center locates in the pillar. Formally, we denote the pillar set of the LiDAR scans U_i as $\mathcal{S}_i = \{u_i^{jk}, i \in \{0,1\}, j \in [0,M), k \in [0,M), u_i^{jk} \in \mathbb{R}^{N \times C}\}$ and u_i^{jk} denotes the point clouds of the pillar on j^{th} row and k^{th} column of the i^{th} LiDAR scan. Considering that the size of foreground objects is small, we use a small pillar size to split the scene, which can truncate more foreground objects as more as possible. With pillars of two LiDAR scans, we can mix them to generate a new LiDAR scan. An intuitive mixing strategy is randomly selecting some pillars from each LiDAR scan and splicing them. However, it is sub-optimal. To make a strong augmentation, we want to leverage each truncated edge of each pillar. Therefore, we alternately select pillars from two LiDAR scans, ensuring adjacent pillars always come from different frames, as shown in Figure 1. Formally, we get a mixed pillar set $\mathcal{S}_{\text{mix}} = \{u_{\text{mix}}^{jk}, j \in [0,M), k \in [0,M)\}$ as follows:

$$\mathcal{S}_{\text{mix}} = \text{PillarMix}(\mathcal{S}_0, \mathcal{S}_1), \tag{2}$$

$$u_{\text{mix}}^{jk} = \begin{cases} u_0^{jk}, & j+k \equiv 0 \pmod 2, \\ u_1^{jk}, & j+k \equiv 1 \pmod 2, \end{cases} \tag{3}$$

Then we concatenate $\mathcal{S}_{\text{mix}}$, resulting in the mixed LiDAR scan U_{mix}. Note that the excessive truncation of point clouds can deteriorate the training since the severely truncated objects are difficult to learn and could confuse the model. Therefore, the pillar size can not be too small.

(Kong et al. 2023; Xiao et al. 2022) are LiDAR segmentation data augmentations that fuse laser beams or polars from different LiDAR scans. Nevertheless, they are incompatible with 3D object detection. On the one hand, their partition strategy is coarse. Considering that detection is an object-aware task and the objects are usually small, we need to design a dense partition strategy so that as many objects as possible can be truncated. On the other hand, the distance-biased partition strategy they use makes the further areas more difficult to truncate and the closer areas to be heavily truncated , which results in insufficient or excessive data augmentation in these areas. Our PillarMix is tailored for 3D object detection, which uses a dense and evenly grid partition strategy, augmenting point clouds of various distances strongly, fully and uniformly. We provide a detailed ablation study in Table 10 to verify the superiority of our PillarMix over other related methods.

Experiments

Datasets and Evaluation Metrics

Waymo Open Dataset Waymo (Sun et al. 2020) is a large-scale LiDAR point cloud dataset, which contains 798 sequences for training and 202 sequences for validation. We

Label Amounts	Method	3D AP/APH @0.7 (LEVEL 2)			
		Overall	Vehicle	Pedestrian	Cyclist
5% ($\sim$ 4k Labels) $\mathcal{P}^L : \mathcal{P}^U = 1:20$	FixMatch (Sohn et al. 2020a)	48.80/43.35	51.87/51.27	48.28/36.56	46.26/42.21
	PseudoAugments† (Leng et al. 2022)	49.73/45.89	54.16/53.61	49.08/39.85	45.94/44.21
	ProficientTeacher (Yin et al. 2022)	51.10/45.75	53.04/52.54	50.33/38.67	49.92/46.03
	PTPM (Ours)	**54.53/51.07**	**56.98/56.48**	**52.64/44.19**	**53.96/52.55**
20% ($\sim$ 16k Labels) $\mathcal{P}^L : \mathcal{P}^U = 1:5$	FixMatch (Sohn et al. 2020a)	55.81/51.45	58.94/58.37	54.37/44.23	54.11/51.75
	PseudoAugments† (Leng et al. 2022)	55.94/52.29	59.55/59.04	56.13/47.04	52.14/50.79
	ProficientTeacher (Yin et al. 2022)	58.59/54.16	59.97/59.36	57.88/46.97	57.93/56.15
	PTPM (Ours)	**60.49/57.02**	**61.62/61.16**	**59.78/51.17**	**60.07/58.74**
100% ($\sim$ 80k Labels) $\mathcal{P}^L : \mathcal{P}^U = 1:1$	FixMatch (Sohn et al. 2020a)	62.06/57.96	63.50/62.98	62.00/52.52	60.69/58.37
	PseudoAugments† (Leng et al. 2022)	61.15/57.75	63.66/63.17	61.37/52.74	58.42/57.34
	ProficientTeacher (Yin et al. 2022)	62.96/59.14	63.56/63.06	62.34/53.19	62.97/61.18
	PTPM (Ours)	**65.73/62.13**	**67.12/66.67**	**65.85/57.11**	**64.23/62.60**

Table 1: Performance on the Waymo Open Dataset with 202 validation sequences. We use the same data split and the same baseline model (SECOND) as ProficientTeacher (Yin et al. 2022). PseudoAugments† is our implementation.

1% Data ($\sim$ 1.4k scenes)	Veh. (LEVEL 1)		Veh. (LEVEL 2)		Ped. (LEVEL 1)		Ped. (LEVEL 2)		Cyc. (LEVEL 1)		Cyc. (LEVEL 2)	
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
PV-RCNN (from DetMatch)	47.3	45.6	43.6	42.0	28.9	15.6	26.2	14.1	-	-	-	-
DetMatch (Park et al. 2022)	52.2	51.1	48.1	47.2	39.5	18.9	35.8	17.1	-	-	-	-
Improvement	+4.9	+5.5	+4.5	+5.2	+10.6	+3.3	+9.6	+3.0	-	-	-	-
PV-RCNN (from HSSDA)	48.5	46.2	45.5	43.3	30.1	15.7	27.3	15.9	4.5	3.0	4.3	2.9
HSSDA (Liu et al. 2023a)	56.4	53.8	49.7	47.3	40.1	20.9	33.5	17.5	29.1	**20.9**	27.9	20.0
Improvement	+7.9	+7.6	+4.2	+4.0	+10.0	+5.2	+6.2	+1.6	+24.6	+17.9	+23.6	+17.1
PV-RCNN (our reproduction)	47.7	38.3	44.1	33.2	27.3	14.1	22.8	11.8	5.5	4.3	5.1	3.4
PTPM (Ours)	**61.5**	**59.8**	**53.7**	**52.2**	**43.1**	**22.3**	**36.3**	**18.8**	**35.7**	17.9	**35.7**	**34.3**
Improvement	+13.8	+21.5	+9.6	+19.0	+15.8	+8.2	+13.5	+7.0	+30.2	+13.6	+30.6	+30.9

Table 2: Performance comparison on the Waymo Open Dataset with 202 validation sequences for the 3D object detection with PV-RCNN (Shi et al. 2020) as the baseline model.

follow ProficientTeachers (Yin et al. 2022) to divide the 798 training sequences equally into labeled split $\mathcal{P}^L$ and unlabeled split $\mathcal{P}^U$, with each containing 399 sequences. Then 5%, 20% and 100% sequences are randomly sampled from $\mathcal{P}^L$, leading to the ratio of labeled data and unlabeled data $\mathcal{P}^L : \mathcal{P}^U$ as 1:20, 1:5 and 1:1, respectively. For a fair comparison, we use the same sequence ids of 5%, 20% and 100% split as ProficientTeachers (Yin et al. 2022).

ONCE Dataset ONCE (Mao et al. 2021) is a large-scale autonomous driving dataset with 1 million LiDAR point cloud samples. There are 15k labeled samples, which are divided into 5K for training, 3k for validation and 8k for testing. The unlabeled samples are divided into 3 subsets: Small ($\sim$100k samples), Medium ($\sim$500k samples) and Large ($\sim$1M samples) to explore the effects of different data amounts for SSL 3D object detection.

Implementation Details

We use SECOND (Yan, Mao, and Li 2018) implemented by OpenPCDet (Team 2020) as our baseline detector, following ProficientTeachers (Yin et al. 2022). If not specified, the ablation studies are conducted under the setting of 5% labeled Waymo dataset. For the training of PatchTeacher, each mini-batch consists of 2 patches of labeled point clouds and 2 patches of unlabeled point clouds. We divide the full point clouds into 4×4 patches. The voxel size of each patch is set to [3.5 cm, 3.5 cm, 3.5 cm]. PatchTeacher is trained for 240 epochs. For the training of the student, each mini-batch consists of 1 frame of labeled point clouds and 4 frames of unlabeled point clouds. The student is trained for 30 epochs. We develop an improved version of PseudoAugment (Leng et al. 2022), semi-sampling, to increase the diversity of training data. Concretely, semi-sampling is a general object sampling strategy, which is a superset of gt-sampling or PseudoAugment. Compared to PseudoAugment, semi-sampling adds the feature of pasting samples cropped from an unlabeled frame to another unlabeled frame, which is more effective than pasting samples cropped from unlabeled frames to labeled frames or from labeled frames to unlabeled frames. More details are provided in the supplementary material.

Comparison with State-of-the-Arts

We perform the comparative study for semi-supervised 3D object detection based on the SECOND and PV-RCNN on the Waymo dataset. The results are presented in Table 1 and Table 2. As depicted in Table 1, our method outperforms state-of-the-art methods by a large margin under all experimental settings of the Waymo dataset. Specifically, for the 5% protocol, we improve mAP from ProficientTeacher's 51.10 to 54.53, which achieves 3.43 mAP improvement. Our PTPM also brings significant improvement when there are more labeled data: 58.59 to 60.49 on the 20% protocol, 62.96 to 65.73 on the 100% protocol, which demonstrates the effectiveness of our method. To further evaluate our method, we use PV-RCNN as our baseline detector and compare the results with DetMatch (Park et al. 2022)

Methods	Vehicle AP (%)				Pedestrian AP (%)				Cyclist AP (%)				mAP (%)
	overall	0-30m	30-50m	50m-inf	overall	0-30m	30-50m	50m-inf	overall	0-30m	30-50m	50m-inf	
Baseline	71.19	84.04	63.02	47.25	26.44	29.33	24.05	18.05	58.04	69.96	52.43	34.61	51.89
Small (100K unlabeled Samples)													
3DIoUMatch	73.81	84.61	68.11	54.48	30.86	35.87	25.55	18.30	56.77	68.02	51.80	35.91	53.81
MeanTeacher	74.46	86.65	68.44	53.59	30.54	34.24	26.31	20.12	61.02	72.51	55.24	39.11	55.34
ProficientTeacher	76.07	86.78	70.19	56.17	35.90	39.98	31.67	24.37	61.19	73.97	55.13	36.98	57.72
PTPM (Ours)	76.27	86.55	69.61	56.02	44.29	51.95	35.86	20.91	61.70	75.19	54.92	34.57	**60.75**
Medium (500K unlabeled Samples)													
3DIoUMatch	75.69	86.46	70.22	56.06	34.14	38.84	29.19	19.62	58.93	69.08	54.16	38.87	56.25
MeanTeacher	76.01	86.47	70.34	55.92	35.58	40.86	30.44	19.82	63.21	74.89	56.77	40.29	58.27
ProficientTeacher	78.07	87.43	72.50	59.51	38.38	42.45	34.62	25.58	63.23	74.70	58.19	40.73	59.89
PTPM (Ours)	76.66	86.75	71.30	56.87	45.87	54.98	37.35	20.89	61.88	74.08	56.52	33.30	**61.47**
Large (1M unlabeled Samples)													
3DIoUMatch	75.81	86.11	71.82	57.84	35.70	40.68	30.34	21.15	59.69	70.69	54.92	39.08	57.07
MeanTeacher	76.38	86.45	70.99	57.48	35.95	41.76	29.05	18.81	65.50	75.72	60.07	43.66	59.28
ProficientTeacher	78.12	87.22	72.74	59.58	41.95	48.09	35.13	26.01	64.12	75.85	58.04	41.45	61.40
PTPM (Ours)	76.46	86.35	71.31	57.08	45.72	55.00	36.81	20.25	65.87	77.41	59.85	42.39	**62.68**

Table 3: Evaluations on ONCE validation set with different amounts of unlabeled samples. We compare our PTPM with 3DIoUMatch (Wang et al. 2021a), MeanTeacher (Tarvainen and Valpola 2017) and ProficientTeacher (Yin et al. 2022).

Exp	Pseudo-L	Semi-S	PillarMix	PatchTeacher	3D AP/APH @0.7 (LEVEL 2)			
					Overall	Vehicle	Pedestrian	Cyclist
(a)					44.87/40.18	49.62/48.99	45.08/34.69	39.91/36.87
(b)	✓				48.63/43.53	53.14/52.56	49.31/38.52	43.44/39.52
(c)	✓	✓			51.61/47.43	54.35/53.81	50.65/40.83	49.84/47.66
(d)	✓	✓	✓		52.88/48.75	55.53/54.98	51.68/41.93	51.42/49.33
(e)	✓	✓	✓	✓	55.81/52.17	57.58/57.07	54.85/46.23	55.00/53.20

Table 4: Ablation study of each component of our method with Waymo 5% labeled dataset. "Pseudo-L" and "Semi-S" represent Pseudo-Labeling and Semi-Sampling, respectively.

and HSSDA (Liu et al. 2023a). As shown in Table 2, our method also achieves considerable improvement over previous methods. In addition, we compare our method with existing SOTA methods on the ONCE dataset. The results in Table 3 show that our approach also exhibits superior results than other methods over all splits. Specifically, our method surpasses ProficientTeacer, by 3.03, 1.58, and 1.28 mAP on small, medium and large splits, once again verifying the advantage of our approaches. For the ONCE dataset, the improvement to the baseline on the pedestrian is the most, which can be attributed to the high-resolution PatchTeacher.

Ablation Study

To better understand how the proposed approach works, we conduct a series of ablation studies under the Waymo 5% labeled data protocol.

Effect of PatchTeacher and PillarMix As shown in table 4, experiment (a) is the baseline which is only supervised by labeled data. Experiment (b) uses pseudo-labeling for semi-supervised learning and experiment (c) employs the semi-sampling strategy (refer to implementation details) to further boost the performance. Comparing experiment (c) with Table 1, we can find that only using pseudo-labeling and semi-sampling can achieve better results than ProficientTeacher. Based on the high-performance experiment (c), our PillarMix further achieves more than 1 mAP improvement in all classes, as shown in experiment (d). Moreover, experiment (e) shows that PatchTeacher can yield substantial improvement over experiment (d).

Exp	P.N.	Q.A.	F.S.	SSL	3D AP/APH @0.7 (LEVEL 2)		
					Vehicle	Pedestrian	Cyclist
(a)					51.83 / 51.26	54.24 / 45.15	49.14 / 46.99
(b)	✓				52.61 / 52.09	54.74 / 45.95	49.36 / 46.96
(c)	✓	✓			55.41 / 54.89	56.77 / 49.75	49.58 / 47.59
(e)	✓	✓	✓		56.51 / 56.00	57.21 / 50.18	49.92 / 48.05
(d)	✓	✓	✓	✓	**58.22 / 57.75**	**57.81 / 50.20**	**57.61 / 56.20**

Table 5: Effect of different components of PatchTeacher. "P.N.", "Q.A.", "F.S." and "SSL" represent Patch Normalizer, Quadrant Align, Fovea Selection and Semi-Supervised Learning, respectively.

Effect of each component of PatchTeacher Here we explore the effect of three key techniques devised for PatchTeacher. As shown in Table 5, experiment (a) is the baseline of PatchTeacher, which uses vanilla partial scene detection. Experiments (b), (c) and (d) utilize Patch Normalizer, Quadrant Align and Fovea Selection, respectively. From the results, we conclude that each component is beneficial for performance, and vehicle and pedestrian classes are improved the most. To further improve the performance, we utilize SSL on our PatchTeacher. Concretely, we leverage pseudo-labeling, semi-sampling and PillarMix to train PatchTeacher like the student. From the comparison between experiments (d) and (e), we note that the SSL contributes the most to the cyclist because the amount of cyclists is small, and its performance is not saturated until we leverage a large number of pseudo labels of the cyclist in unlabeled data. Eventually, we improve the simple partial scene detection by absolutely 6.39, 3.57 and 8.47 mAP on vehicle, pedestrian and cyclist, respectively. The improvement to the com-

plete scene detection (the student) is even more, comparing experiment (d) of Table 5 and experiment (a) of Table 4. We also evaluate the pseudo labels generated by the student and PatchTeacher on the *unlabeled split*, as shown in Table 6. The results show that our PatchTeacher can achieve more than 10 mAPH improvement over the student, which verifies the rationality of our method.

Model	3D AP/APH @0.7 (LEVEL 2)			
	Overall	Vehicle	Pedestrian	Cyclist
Student	46.20 / 41.40	51.39 / 50.75	47.11 / 36.06	40.09 / 37.40
Teacher	56.53 / 53.49	57.74 / 57.25	60.29 / 53.23	51.56 / 49.98

Table 6: Evaluation results of the student and teacher (PatchTeacher) on the Waymo unlabeled split.

Voxel (m)	3D AP/APH @0.7 (LEVEL 2)			
	Overall	Vehicle	Pedestrian	Cyclist
0.100	46.21 / 41.92	54.30 / 53.71	43.02 / 33.13	41.30 / 38.93
0.050	52.85 / 49.92	55.39 / 55.45	54.16 / 46.70	49.01 / 47.60
0.040	53.86 / 50.78	56.17 / 55.66	56.41 / 48.94	49.00 / 47.75
0.035	**54.38 / 51.41**	56.51 / 56.00	57.21 / 50.18	49.42 / 48.05
0.003	54.23 / 51.30	56.31 / 55.81	57.10 / 50.07	49.27 / 48.02

Table 7: Ablation study of voxel size of PatchTeacher.

α	3D AP/APH @0.7 (LEVEL 2)			
	Overall	Vehicle	Pedestrian	Cyclist
1	57.64 / 54.46	58.10 / 57.53	57.74 / 50.04	57.09 / 55.81
2	57.55 / 54.45	58.13 / 57.65	57.73 / 50.12	56.80 / 55.58
3	**57.88 / 54.72**	58.22 / 57.75	57.81 / 50.20	57.61 / 56.20
4	57.83 / 54.72	58.35 / 57.89	57.66 / 50.18	57.47 / 56.08
Avg	48.64 / 46.33	55.43 / 55.06	44.83 / 39.18	45.65 / 44.74

Table 8: Ablation study of Patch Normalizer α. "Avg" represents that we average the negative classification loss by the number of negative samples.

Effect of hyperparameters We now investigate the effect of different hyperparameters in our framework. We first ablate the voxel size of PatchTeacher. As shown in Table 7, we observe the best results are achieved when the voxel size is set to [3.5cm, 3.5cm, 3.5cm], about 1/3 of the student that uses a voxel size of [10cm, 10cm, 15cm]. Directly using a small voxel size for voxelization on a complete scene will increase a huge amount of memory consumption, which is unaffordable. This emphasizes the importance of partial scene detection. We then evaluate the effect of Patch Normalizer factor α. The results are shown in Table 8. When $\alpha = 3$, the model produces the highest mAP. We also conduct an experiment that averages negative classification loss by the number of negative samples. As shown in Table 8, the performance drops drastically. Then, we explore how the pillar size of PillarMix affects the performance of SSL in Table 9. Setting the pillar size too large can not make a strong edge-truncated augmentation, while setting it too small may produce over-difficult samples, which is harmful to optimization. When using a pillar size of 5 meters, the model can achieve the best performance.

Comparison with PillarMix and other approaches To verify the superiority of our PillarMix over other relative approaches, we conduct a comparative experiment, as shown in Table 10. We apply the LaserMix, PolarMix, Shuffle Data Augmentation and our PillarMix on the baseline, which only uses pseudo labeling for SSL. Here shuffle data augmentation is proposed by (Liu et al. 2023a), which permutes the patches of a LiDAR scan, while it may pull in distant patches

Pillar (m)	3D AP/APH @0.7 (LEVEL 2)			
	Overall	Vehicle	Pedestrian	Cyclist
2.5	52.67 / 48.60	55.68 / 55.14	51.67 / 41.88	50.67 / 48.78
5	**52.77 / 48.60**	55.44 / 54.87	51.82 / 42.03	51.04 / 48.91
10	52.29 / 48.09	55.16 / 54.59	51.66 / 41.63	50.04 / 48.06
20	52.34 / 48.14	54.83 / 54.27	51.26 / 41.33	50.92 / 48.82

Table 9: Ablation study of pillar size of PillarMix.

Method	3D AP/APH @0.7 (LEVEL 2)		
	Vehicle	Pedestrian	Cyclist
Baseline	54.35 / 53.81	50.85 / 40.83	49.84 / 47.66
PolarMix	55.03 / 54.19	51.05 / 41.33	50.04 / 48.05
LaserMix	55.06 / 54.34	51.19 / 41.14	49.39 / 47.37
ShuffleDA	54.96 / 54.23	50.97 / 40.88	50.19 / 48.00
PillarMix	**55.44 / 54.87**	**51.82 / 42.03**	**51.04 / 48.91**

Table 10: Comparsion of PillarMix and other data augmentation approaches: PolarMix (Xiao et al. 2022), LaserMix (Kong et al. 2023) and ShuffleDA (Liu et al. 2023a).

and push away close point clouds, which destroys the intrinsic characteristic of LiDAR point clouds. The results show that each approach can improve the baseline, while our PillarMix exhibits better results, which can be attributed to the sufficient and uniform partition of PillarMix.

Partial scene detection on different detectors Partial scene detection can break the memory bottleneck to leverage a extremely small voxel size for feature extraction. Here we explore partial scene performances of three mainstream detectors. As shown in Figure 11, when using partial scene detection, we achieve significant improvement over the complete scene detection with different detectors.

Method	Partial Scene Detection	3D AP/APH @0.7 (LEVEL 2)		
		Vehicle	Pedestrian	Cyclist
SECOND		49.62/48.99	45.08/34.69	39.91/36.87
	✓	**56.51/56.00**	**57.21/50.18**	**49.42/48.05**
PV-RCNN		5705/56.15	52.09/29.07	48.75/44.51
	✓	**59.61/58.98**	**60.48/53.07**	**55.50/53.76**
CenterPoint		48.21/47.62	50.10/43.79	44.36/43.10
	✓	**57.22/56.77**	**60.31/54.74**	**47.19/46.24**

Table 11: Performance of different detectors with partial scene detection with 5% labeled data (no unlabeled data).

Conclusion

In this paper, we aim to improve the performance of semi-supervised 3D object detection. We propose PatchTeacher, which leverages the low memory cost advantage of partial scene detection to equip a high-resolution voxelization. The PatchTeacher can generate strong pseudo labels for students, which can greatly improve semi-supervised learning. Moreover, we present a strong data augmentation, PillarMix, to further boost the performance of our SSL framework. Our method sets a new state of the art for semi-supervised 3D object detection in both Waymo and ONCE datasets.

Acknowledgements

This work was supported in part by The National Nature Science Foundation of China (Grant Nos: 62273301, 62273302, 62036009, 61936006, 62273303), in part by Ningbo Key RD Program (No.2023Z231, 2023Z229), in part by Yongjiang Talent Introduction Programme (Grant No: 2022A-240-G), in part by the Key RD Program of Zhejiang Province, China (2023C01135), in part by the National Key RD Program of China (NO.2022ZD0160101).

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Bachman, P.; Alsharif, O.; and Precup, D. 2014. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27.

Chen, X.; Shi, S.; Zhu, B.; Cheung, K. C.; Xu, H.; and Li, H. 2022. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *European Conference on Computer Vision*, 680–697. Springer.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5070–5079.

Jeong, J.; Lee, S.; Kim, J.; and Kwak, N. 2019. Consistency-based semi-supervised learning for object detection. In *Advances in neural information processing systems*, 10759–10768.

Kong, L.; Ren, J.; Pan, L.; and Liu, Z. 2023. Lasermix for semi-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21705–21715.

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896. Atlanta.

Leng, Z.; Cheng, S.; Caine, B.; Wang, W.; Zhang, X.; Shlens, J.; Tan, M.; and Anguelov, D. 2022. PseudoAugment: Learning to Use Unlabeled Data for Data Augmentation in Point Clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, 555–572. Springer.

Li, X.; Ma, T.; Hou, Y.; Shi, B.; Yang, Y.; Liu, Y.; Wu, X.; Chen, Q.; Li, Y.; Qiao, Y.; et al. 2023. LoGoNet: Towards Accurate 3D Object Detection with Local-to-Global Cross-Modal Fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17524–17534.

Liu, C.; Gao, C.; Liu, F.; Li, P.; Meng, D.; and Gao, X. 2023a. Hierarchical Supervision and Shuffle Data Augmentation for 3D Semi-Supervised Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23819–23828.

Liu, Y.-C.; Ma, C.-Y.; He, Z.; Kuo, C.-W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; and Vajda, P. 2021. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023b. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2774–2781. IEEE.

Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; et al. 2021. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*.

Park, J.; Xu, C.; Zhou, Y.; Tomizuka, M.; and Zhan, W. 2022. DetMatch: Two Teachers are Better Than One for Joint 2D and 3D Semi-Supervised Object Detection. *arXiv preprint arXiv:2203.09510*.

Peng, L.; Liu, F.; Yu, Z.; Yan, S.; Deng, D.; Yang, Z.; Liu, H.; and Cai, D. 2022. Lidar point cloud guided monocular 3d object detection. In *European Conference on Computer Vision*, 123–139. Springer.

Qi, C. R.; Zhou, Y.; Najibi, M.; Sun, P.; Vo, K.; Deng, B.; and Anguelov, D. 2021. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6134–6144.

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, 1163–1171.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10529–10538.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020a. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.

Sohn, K.; Zhang, Z.; Li, C.-L.; Zhang, H.; Lee, C.-Y.; and Pfister, T. 2020b. A Simple Semi-Supervised Learning Framework for Object Detection. *arXiv preprint arXiv:2005.04757*.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Tang, P.; Ramaiah, C.; Wang, Y.; Xu, R.; and Xiong, C. 2021. Proposal learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2291–2301.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets im-

prove semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Team, O. D. 2020. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet.

Wang, H.; Cong, Y.; Litany, O.; Gao, Y.; and Guibas, L. J. 2021a. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14615–14624.

Wang, J.; Gang, H.; Ancha, S.; Chen, Y.-T.; and Held, D. 2021b. Semi-supervised 3D object detection via temporal graph neural networks. In *2021 International Conference on 3D Vision (3DV)*, 413–422. IEEE.

Wu, X.; Peng, L.; Yang, H.; Xie, L.; Huang, C.; Deng, C.; Liu, H.; and Cai, D. 2022. Sparse Fuse Dense: Towards High Quality 3D Detection with Depth Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5418–5427.

Xiao, A.; Huang, J.; Guan, D.; Cui, K.; Lu, S.; and Shao, L. 2022. Polarmix: A general data augmentation technique for lidar point clouds. *Advances in Neural Information Processing Systems*, 35: 11035–11048.

Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; and Liu, Z. 2021. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3060–3069.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yin, J.; Fang, J.; Zhou, D.; Zhang, L.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022. Semi-supervised 3D object detection with proficient teachers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, 727–743. Springer.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

Zhao, N.; Chua, T.-S.; and Lee, G. H. 2020. Sess: Self-ensembling semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11079–11087.