PointDistiller: Structured Knowledge Distillation Towards Efficient and Compact 3D Detection

Linfeng Zhang\textsuperscript{1}\textdagger\textdagger, Runpei Dong\textsuperscript{2}\textdagger, Hung-Shuo Tai\textsuperscript{3}, Kaisheng Ma\textsuperscript{1}
Tsinghua University\textsuperscript{1} Xi’an Jiaotong University\textsuperscript{2} DIDI\textsuperscript{3}

Abstract

The remarkable breakthroughs in point cloud representation learning have boosted their usage in real-world applications such as self-driving cars and virtual reality. However, these applications usually have a strict requirement for not only accurate but also efficient 3D object detection. Recently, knowledge distillation has been proposed as an effective model compression technique, which transfers the knowledge from an over-parameterized teacher to a lightweight student and achieves consistent effectiveness in 2D vision. However, due to point clouds’ sparsity and irregularity, directly applying previous image-based knowledge distillation methods to point cloud detectors usually leads to unsatisfactory performance. To fill the gap, this paper proposes PointDistiller, a structured knowledge distillation framework for point clouds-based 3D detection. Concretely, PointDistiller includes local distillation which extracts and distills the local geometric structure of point clouds with dynamic graph convolution and reweighted learning strategy, which highlights student learning on the crucial points or voxels to improve knowledge distillation efficiency. Extensive experiments on both voxels-based and raw points-based detectors have demonstrated the effectiveness of our method over seven previous knowledge distillation methods. For instance, our 4× compressed PointPillars student achieves 2.8 and 3.4 mAP improvements on BEV and 3D object detection, outperforming its teacher by 0.9 and 1.8 mAP, respectively. Codes are available in https://github.com/RunpeiDong/PointDistiller.

1. Introduction

The growth in large-scale lidar datasets\textsuperscript{14} and the achievements in end-to-end 3D representation learning\textsuperscript{46, 47} have boosted the developments of point cloud based segmentation, generation, and detection\textsuperscript{25, 48}. As one of the essential tasks of 3D computer vision, 3D object detection plays a fundamental role in real-world applications such as autonomous driving cars\textsuperscript{3, 6, 14} and virtual reality\textsuperscript{43}. However, recent research has shown a growing discrepancy between cumbersome 3D detectors that achieve state-of-the-art performance and lightweight 3D detectors which are affordable in real-time applications on edge devices. To address this problem, sufficient model compression techniques have been proposed, such as network pruning\textsuperscript{18, 35, 37, 73}, quantization\textsuperscript{8, 12, 40}, lightweight model design\textsuperscript{21, 38, 51}, and knowledge distillation\textsuperscript{20}.

Knowledge distillation, which aims to improve the performance of a lightweight student model by training it to mimic a pre-trained and over-parameterized teacher model, has evolved into one of the most popular and effective model compression methods in both computer vision and natural language processing\textsuperscript{20, 50, 52, 66}. Sufficient theoretical and empirical results have demonstrated its effectiveness in image-based visual tasks such as image classification\textsuperscript{20, 50}, semantic segmentation\textsuperscript{33} and object detection\textsuperscript{1, 5, 28, 71}. However, compared with images, point clouds have their properties: (i) Point clouds inherently lack topological information, which makes recovering the local topology information crucial for the visual tasks\textsuperscript{26, 39, 65}. (ii) Different from images that have a regular structure, point clouds are irregularly and sparsely distributed in the metric space\textsuperscript{13, 15}.

These differences between images and point clouds have hindered the image-based knowledge distillation methods from achieving satisfactory performance on point clouds and also raised the requirement to design specific knowledge distillation methods for point clouds. Recently, a few methods have been proposed to apply knowledge distillation to 3D
we propose a reweighted learning strategy, which highlights which contain multiple points. Motivated by this observation, we propose a structured knowledge distillation framework.

Local Distillation

Sufficient recent studies show that capturing and making use of the semantic information in the local geometric structure of point clouds have a crucial impact on point cloud representation learning [47, 64]. Hence, instead of directly distilling the backbone feature of teacher detectors to student detectors, we propose local distillation, which firstly clusters the local neighboring voxels or points with KNN (K-Nearest Neighbours), then encodes the semantic information in local geometric structure with dynamic graph convolutional layers [64], and finally distill them from teachers to students. Hence, the student detectors can inherit the teacher’s ability to understand point clouds’ local geometric information and achieve better detection performance.

Reweighted Learning Strategy

One of the mainstream methods for processing point clouds is to convert them into volumetric voxels and then encode them as regular data. However, due to the sparsity and the noise in point clouds, most of these voxels contain only a single point. For instance, as shown in Figure 2, on the KITTI dataset, around 68% voxels in point clouds contain only one point, which has a high probability of being a noise point. Hence, the representative features in these single-point voxels have relatively lower importance in knowledge distillation compared with the voxels which contain multiple points. Motivated by this observation, we propose a reweighted learning strategy, which highlights student learning on the voxels with multiple points by giving them larger learning weights. Besides, the similar idea can also be easily extended to raw points-based detectors to highlight knowledge distillation on the points with more considerable influence on the prediction.

Extensive experiments on both voxels-based and raw-points based detectors have been conducted to demonstrate the effectiveness of our method over the previous seven knowledge distillation methods. As shown in Figure 1, on PointPillars and SECOND detectors, our method leads to 4× compression and 0.9~1.8 mAP improvements at the same time. On PointRCNN, our method leads to 8× compression with only 0.2 BEV mAP drop. Our main contributions be summarized as follows.

- We propose local distillation, which firstly encodes the local geometric structure of point clouds with dynamic graph convolution and then distills them to students.
- We propose reweighted learning strategy to handle the sparsity and noise in point clouds. It highlights student learning on the voxels, which have more points inside them, by giving them higher learning weights in knowledge distillation.
- Extensive experiments on both voxels-based and raw points-based detectors have been conducted to demonstrate the performance of our method over seven previous methods. Besides, we have released our codes to promote future research.

2. Related Work

2.1. Knowledge Distillation

The idea of training a small model with a large pre-trained model was firstly proposed by Bucilă et al. for ensemble model compression [2]. Then, with the excellent breakthroughs of deep learning, Hinton et al. propose the concept of knowledge distillation which strives to compress an over-parameterized teacher model by transferring its knowledge to a lightweight student model [20]. Early knowledge distillation methods usually train the students to mimic the predicted categorical probability distribution of teachers [20, 74]. Then, extensive methods have been proposed to learn teacher knowledge in the backbone features [50] or its variants, such as attention [69, 71], relation [31, 42, 45, 60], task-oriented information [72] and so on. Following its success in classification, abundant works have applied knowledge distillation to object detection [5, 28, 61, 71], segmentation [33], image generation [22, 27, 29, 31, 49, 70], pre-trained language models [52, 66], semi-supervised learning [24, 58] and lead to consistent effectiveness.

Knowledge Distillation on Object Detection

Recently, designing specific knowledge distillation methods to improve the efficiency and accuracy of object detection has become a rising and popular topic. Chen et al. first propose to apply the naive prediction and feature-based knowledge distillation methods to object detection [5]. Then, Wang et al. show that the imbalance between foreground objects
and background objects hinders knowledge distillation from achieving better performance in object detection [61]. To address this problem, abundant knowledge distillation methods have tried to find the to-be-distilled regions based on the ground-truth [61], detection results [11], spatial attention [71], query-based attention [23] and gradients [16]. Moreover, recent methods have also been proposed to distill the pixel-level and object-level relation from teachers to students [11, 34, 71]. Besides knowledge distillation for 2D detection, some cross-modal knowledge distillation have been introduced to transfer knowledge from RGB-based teacher detectors to lidar-based student detectors or vice versa [9, 17, 53]. However, most of these methods focus on the choice of students and teachers in a multi-modal framework, while the design of specific knowledge distillation optimization methods on point clouds based pure 3D detection has not been well-explored.

2.2. 3D Object Detection on Point Clouds

The rapid development of deep learning has firstly boosted the research in 2D object detection and then recently increased the research trend in point clouds-based 3D object detection. PointNet [46] is firstly proposed to extract the feature of points with multi-layer perception in an end-to-end manner. Then, PointNet++ is further proposed to capture the local structures in a hierarchical fashion with density adaptive sampling and grouping [47]. Zhou et al. propose VoxelNet, a single-stage detector that divides a point cloud into equally spaced 3D voxels and processes them with voxel feature encoding layers [77]. Then, SECOND is proposed to improve VoxelNet with sparse convolutional layers and focal loss [67]. PointPillars is proposed to divide point clouds into several pillars and then convert them into a pseudo image, which can be further processed with 2D convolutional layers [25, 41]. Shi et al. propose PointRCNN, a two-stage detection method that firstly generates bottom-up 3D proposals based on the raw point clouds and then refines them to obtain the final detection results [55]. Afterward, Fast Point R-CNN and PV-RCNN are proposed to utilize both voxel representation and raw point clouds to exploit their respective advantages [7, 54]. Recently, Qi et al. propose to perform offboard 3D detection with point cloud sequences, which is able to make use of the temporal points and achieve comparable performance with human labels [48]. The graph convolutional neural network is another rising star in point cloud detection [56, 64]. Following [4], Wang and Solomon propose to model 3D detection as graph message passing with set-to-set prediction, which removes post-processing necessity [63]. Zhou et al. propose adaptive graph convolution, which generates adaptive kernels according to the learned features [76].

Efficient 3D Object Detectors

Unfortunately, the significant 3D detection performance usually comes at the expense of high computational and storage costs, making them unaffordable in real-time applications such as self-driving cars. To address this issue, recent research attention has been paid to designing efficient 3D detectors. Tang et al. propose to apply neural architecture search to 3D detection by using sparse point-voxel convolution [57]. Li et al. propose Lidar-RCNN, which resorts to a point-based approach and remedies the problem of uncorrected proposal sizes [32]. Liu et al. propose voxel-point cnn to represent the 3D input data in points while performing the convolutions in voxels to reduce the memory accessing consumption [36]. Recently, Li et al. propose to improve the efficiency of graph convolution for point clouds by simplified KNN and graph shuffling [30].

3. Methodology

3.1. Preliminaries

Given a set of point clouds $X = \{x_1, x_2, ..., x_n\}$ and the corresponding label set $Y = \{y_1, y_2, ..., y_m\}$, the object detector can be formulated as $F = f \circ g$, where $f$ is the feature encoding layer to extract representation features from inputs and $g$ is the detection head for prediction. Then, the representation feature on the sample $x$ can be written as $f(x) \in \mathbb{R}^{n \times C}$, where $n$ indicates the number of voxels for voxels-based detectors or the number of points for raw points-based detectors. $C$ indicates the number of channels. Besides, for voxels-based detectors, we define $v_{ij}(x) = 1$ if the $j$-th point of $x$ belongs to the $i$-voxel else 0. Then, the number of points in the $i$-th voxel can be denoted as $\sum_j v_{ij}(x)$. Usually, knowledge distillation involves a to-be-trained student detector and a pre-trained teacher detector, and we distinguish them with scripts $S$ and $T$, respectively.

3.2. Our Method

Sampling Top-$N$ To-be-distilled Voxels (Points)

As discussed in previous sections, since the point clouds are overwhelmingly sparse while the voxels are usually equally spaced, most of the voxels only contain very few and even single point. Thus, these single-point voxels have much less value to be learned by students in knowledge distillation. Even in raw points-based detectors, there usually exist some points which are relatively more crucial and some points which are not meaningful (e.g., the noise points). Thus, instead of distilling all the voxels or points in point clouds, we propose to distill the voxels or points which are more valuable for knowledge distillation. Concretely, for voxels-based detectors, we define the importance score of $i$-th voxel as $\sum_j v_{ij}(x)$, which indicates the number of points inside it. For point-based detectors, motivated by previous works which localized the crucial pixels in images with attention, we define the importance score for $i$-th point as its permutation-invariant maximal value along the channel dimension, which can be formulated as $\max \{f(x)[i]\}$. Based
Figure 3. The overview of our method. $f^T$ and $f^S$: the feature encoding layers in the teacher and student detectors. $A^T$ and $A^S$: features of the sampled to-be-distilled voxels or points with top-$N$ largest importance score. $C_T$ and $C_S$: the number of channels for features of the teacher and the student detectors. $G_T$ and $G_S$: the graph features of the teacher and student detectors. Based on the pre-defined importance score, our method samples the relatively more crucial $N$ voxels or points from the whole point cloud, extracts their local geometric structure of them with dynamic graph convolution, and then distills them in a reweighted manner.

Extracting Local Geometric Information As pointed out by abundant previous works, the local geometric information has a crucial influence on the performance of point cloud detectors [47, 64]. Thus, instead of directly distilling the representative feature, we propose local distillation which extracts the local geometric information of point clouds with dynamic graph convolution layers and distills it to the student detector. Concretely, denoting $z_i = A(x)[i]$ as the feature of the $i$-th to-be-distilled voxel or point, we can build a graph based on this voxel or point and its $K$ neighboring voxels or points clustered by KNN (K-Nearest Neighbours). By denoting the features of $z_i$ and its $K - 1$ neighbours as $z_{i,1}$ and $N_i = \{z_{i,2}, z_{i,3}, ..., z_{i,K}\}$ respectively, motivated by previous methods [46, 64], we firstly update the feature of each voxel (or point) in this graph by concatenating them with the global centroid voxel (or point) feature $z_{i,1}$, which can be formulated as $\hat{z}_{i,j} = \text{cat}(z_{i,1}, z_{i,j})$ for all $z_{i,j} \in N_i$. Then, we apply a dynamic graph convolution as the aggregation operation upon them, which can be formulated as $G_i = \gamma(\hat{z}_{i,1}, ..., \hat{z}_{i,K})$, where $\gamma$ is the aggregation operator. Following previous graph-based point cloud networks, we set $\gamma$ as a nonlinear layer with ReLU activation and batch normalization. Then the training objective of local distillation can be formulated as

$$
\arg \min_{\theta_S, \theta_T} \mathbb{E}_{x} \left[ \frac{1}{N} \sum_{i=1}^{N} \|G_i^S(x) - G_i^T(x)\| \right],
$$

where $\theta_S$ indicates the parameters of student encoding layer $f^S$. $\theta_T = [\theta_{\gamma}, \theta_{\psi}, \theta_{\phi}]$ indicates the parameters of dynamic graph convolution layers for the student and teacher detectors. Note that these layers are trained with the student simultaneously and can be discarded during inference.

Reweighting Knowledge Distillation Loss Usually, compared with the teacher detector, the student detector has much fewer parameters, implying inferior learning capacity. Thus, it is challenging for the student detector to inherit teacher knowledge in all points or voxels. As discussed above, different voxels and points in point cloud object detection have different values in knowledge distillation. Thus, we propose to reweight the learning weight of each voxel or point based on the importance score introduced in previous paragraphs. Denote the learning weight for the $N$ to-be-distilled as $\phi \in \mathbb{R}^N$. 

![Diagram](image-url)
where $\mathbf{w}$ indicates whether our method is utilized. The reported result in the first line of each detector is the performance of the teacher detector.

$$\text{arg min}_{\theta_S, \theta_T} \mathbb{E}_x \left[ \frac{1}{N} \sum_{i=1}^{N} \phi_i \cdot \| G_i^S(x) - G_i^T(x) \| \right].$$

(2)
As shown in the above loss function, with a higher $\phi_i$, the knowledge distillation loss between student and teacher features at the $i$-th graph will have a more extensive influence on the overall loss, and thus student learning on the $i$-th graph can be highlighted. As a result, the proposed reweighting strategy allows the student detector to pay more attention to learning teacher knowledge in the relatively more crucial voxel graphs (point graphs). Moreover, Equation 2 also implies that our method is a feature-based knowledge distillation method that is not correlated with the architecture of detectors and the label set $\mathcal{Y}$. Hence, it can be directly added to the origin training loss of all kinds of 3D object detectors for model compression.

### 4. Experiment

#### 4.1. Experiment Setting

We have evaluated our method in both voxels-based object detector including PointPillars [25], SECOND [67] and CenterPoint [68], and the raw points based object detector including PointRCNN [55]. Most experiments are conducted on KITTI [14] and nuScenes [3], which consist of samples that have both lidar point clouds and images. Our models are trained with only the lidar point clouds. For KITTI, we report the average precision calculated by 40 sampling recall positions for BEV (Bird’s Eye View) object detection and 3D object detection on the validation split. Following the typical protocol, the IoU threshold is set as 0.7 for class Car and 0.5 for class Pedestrians and Cyclists. We have mainly compared our methods with seven previous knowledge distillation methods, including methods proposed by Remero et al. [50], Zagoruko et al. [69], Tung et al. [60], Heo et al. [19], Zheng et al. [75], Tian et al. [59], and Zhang et al. [71]. All the experiments are conducted with mmdetection3d [10] and PyTorch [44]. We keep the training and evaluation settings in mmdetection3d as default. The teacher model is the origin model before compression. The student model shares the same architecture and neural network depth as

<table>
<thead>
<tr>
<th>Task</th>
<th>Method</th>
<th>Car</th>
<th>Pedestrians</th>
<th>Cyclists</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Teacher w/o KD</td>
<td>94.3</td>
<td>88.1</td>
<td>83.6</td>
<td>57.9</td>
</tr>
<tr>
<td></td>
<td>Student w/o KD</td>
<td>92.4</td>
<td>88.2</td>
<td>83.6</td>
<td>53.0</td>
</tr>
<tr>
<td></td>
<td>+ Romero et al. [50]</td>
<td>91.5</td>
<td>85.6</td>
<td>83.1</td>
<td>57.5</td>
</tr>
<tr>
<td></td>
<td>+ Zagoruko et al. [69]</td>
<td>92.6</td>
<td>88.0</td>
<td>83.6</td>
<td>56.7</td>
</tr>
<tr>
<td></td>
<td>+ Zheng et al. [75]</td>
<td>92.7</td>
<td>87.9</td>
<td>83.2</td>
<td>57.7</td>
</tr>
<tr>
<td></td>
<td>+ Tung et al. [60]</td>
<td>92.8</td>
<td>88.0</td>
<td>83.3</td>
<td>54.5</td>
</tr>
<tr>
<td></td>
<td>+ Tian et al. [59]</td>
<td>92.7</td>
<td>87.8</td>
<td>83.2</td>
<td>56.6</td>
</tr>
<tr>
<td></td>
<td>+ Heo et al. [19]</td>
<td>92.6</td>
<td>87.9</td>
<td>83.5</td>
<td>57.6</td>
</tr>
<tr>
<td></td>
<td>+ Zhang et al. [71]</td>
<td>92.3</td>
<td>87.5</td>
<td>83.0</td>
<td>57.9</td>
</tr>
<tr>
<td></td>
<td>+ Ours</td>
<td>93.1</td>
<td>89.0</td>
<td>86.3</td>
<td>59.8</td>
</tr>
</tbody>
</table>

Table 4. Ablation study on 4× compressed PointPillars students. LD and RL indicates local distillation and the reweighted learning strategy, respectively. mAP is measured on the moderate difficulty.

As shown in Table 4, the knowledge distillation loss between student and teacher features at the $i$-th graph will have a more extensive influence on the overall loss, and thus student learning on the $i$-th graph can be highlighted. As a result, the proposed reweighting strategy allows the student detector to pay more attention to learning teacher knowledge in the relatively more crucial voxel graphs (point graphs). Moreover, Equation 2 also implies that our method is a feature-based knowledge distillation method that is not correlated with the architecture of detectors and the label set $\mathcal{Y}$. Hence, it can be directly added to the origin training loss of all kinds of 3D object detectors for model compression.
its teacher but with fewer channels. Note that experiments on students with less neural layers are also provided in the appendix. Following previous works, the average precision of three difficulties and the three categories are reported as the performance metrics [14]. Please refer to our codes in the supplementary material for more details.

### 4.2. Experimental Results

Table 1 shows the performance of detectors trained with and without our method for BEV detection and 3D detection, respectively. It is observed that: (i) Significant average precision improvements on all kinds of detectors and all compression ratios for both BEV and 3D detection. On average, 2.4 and 1.0 moderate mAP improvements can be observed for the voxel and raw points-based detectors, respectively. On BEV and 3D detection, 1.9 and 1.9 moderate mAP improvements can be obtained, respectively. (ii) On the BEV detection of PointPillars and SECOND detectors, the 4× compressed and accelerated students trained with our method outperform their teachers by 0.9 and 0.9 mAP, respectively. On the 3D detection of PointPillars and SECOND detectors, the 4× compressed and accelerated students trained with our method outperform their teachers by 0.9 and 0.9 mAP, respectively. (iii) Consistent average precision boosts can be observed in detection results of all difficulties. For instance, on moderate BEV detection of PointPillars students, 2.4, 2.3, and 2.3 mAP improvements can be observed for easy, moderate, and hard difficulties, respectively. These observations demonstrate that our method can successfully transfer teacher knowledge to the student detectors. (iv) Consistent average precision boosts can be observed in detection results of all categories. For instance, on moderate BEV detection of PointPillars students, 0.6, 3.2 and 3.1 mAP improvements can be obtained.

![Figure 5. A sensitivity study to hyper-parameters in our method on KITTI with 4× compressed PointPillars detectors. mAP is measured on the moderate difficulty.](image-url)

![Figure 6. Visualization on importance scores for PointPillars. Red points indicate the voxels with high importance scores.](image-url)

Table 5. Experimental results on nuScenes dataset with PointPillars and CenterPoint. A higher mAP and NDS is better.

<table>
<thead>
<tr>
<th>Model</th>
<th>FLOPs (G)</th>
<th>#Params (M)</th>
<th>KD Method</th>
<th>mAP(%)</th>
<th>NDS(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointPillars</td>
<td>34.3</td>
<td>4.8</td>
<td>Teacher w/o KD</td>
<td>39.3</td>
<td>53.2</td>
</tr>
<tr>
<td>Student w/o KD</td>
<td>36.0</td>
<td>50.5</td>
<td>Student w/o KD</td>
<td>32.2</td>
<td>57.3</td>
</tr>
<tr>
<td>+ Heo et al. [19]</td>
<td>36.2,+0.2</td>
<td>50.6,+0.1</td>
<td>Student w/o KD</td>
<td>32.4,+0.2</td>
<td>57.4,+0.2</td>
</tr>
<tr>
<td>+ Tian et al. [59]</td>
<td>53.7,+0.3</td>
<td>50.4,+0.1</td>
<td>Student w/o KD</td>
<td>32.3,+0.1</td>
<td>57.5,+0.1</td>
</tr>
<tr>
<td>+ Wang et al. [62]</td>
<td>36.2,+0.2</td>
<td>50.7,+0.2</td>
<td>Ours</td>
<td>36.7,+0.7</td>
<td>51.0,+0.6</td>
</tr>
<tr>
<td>+ Ours</td>
<td>51.0</td>
<td>51.0</td>
<td>Ours</td>
<td>48.6,+1.3</td>
<td>51.0,+1.3</td>
</tr>
<tr>
<td>CenterPoint</td>
<td>121.2</td>
<td>9.2</td>
<td>Teacher w/o KD</td>
<td>57.3</td>
<td>65.6</td>
</tr>
<tr>
<td>Student w/o KD</td>
<td>55.4</td>
<td>64.2</td>
<td>Student w/o KD</td>
<td>55.4</td>
<td>64.2</td>
</tr>
<tr>
<td>+ Heo et al. [19]</td>
<td>55.7,+0.3</td>
<td>64.4,+0.2</td>
<td>Student w/o KD</td>
<td>55.7,+0.3</td>
<td>64.5,+0.3</td>
</tr>
<tr>
<td>+ Tian et al. [59]</td>
<td>55.9,+0.5</td>
<td>64.6,+0.4</td>
<td>Student w/o KD</td>
<td>56.4,+1.0</td>
<td>65.1,+0.9</td>
</tr>
<tr>
<td>+ Wang et al. [62]</td>
<td>55.8,+0.1</td>
<td>64.6,+0.8</td>
<td>Ours</td>
<td>56.4,+1.0</td>
<td>65.1,+0.9</td>
</tr>
<tr>
<td>+ Ours</td>
<td>56.4</td>
<td>65.1</td>
<td>Ours</td>
<td>48.6,+1.0</td>
<td>51.0,+1.0</td>
</tr>
<tr>
<td>+ Ours</td>
<td>57.0</td>
<td>65.3</td>
<td>Ours</td>
<td>48.6,+1.0</td>
<td>51.0,+1.0</td>
</tr>
</tbody>
</table>

Figure 5. A sensitivity study to hyper-parameters in our method on KITTI with 4× compressed PointPillars detectors. mAP is measured on the moderate difficulty.
5. Discussion

5.1. Ablation Study and Sensitivity Study

Ablation Study  The proposed PointDistiller is mainly composed of two components, including the reweighted learning strategy (RL) and local distillation (LD). Ablation studies with $4 \times$ compressed PointPillars students on KITTI are shown in Table 4. It is observed that: (i) 2.0 and 1.9 mAP improvements can be obtained by only using the reweighted learning strategy to distill the backbone features on BEV detection and 3D detection, respectively. (ii) 2.3 and 2.5 mAP boosts can be gained by using local distillation without reweighted learning on BEV detection and 3D detection, respectively. (iii) By combining the two methods together, 0.5 and 0.9 further mAP improvements can be achieved on BEV detection and 3D detection, respectively. These observations indicate that each module in PointDistiller has its individual effectiveness and their merits are orthogonal. Besides, they also implies that the proposed local distillation and reweighted learning may be combined with other knowledge distillation methods to achieve better performance.

Sensitivity Study  Our method mainly introduces two hyper-parameters, $K$ and $N$, indicating the number of nodes in a graph for local distillation and the number of to-be-distilled voxels (points) respectively. A hyper-parameter sensitivity study on them is shown in Figure 5. It is observed that our method with different hyper-parameter values consistently outperforms the baseline by a large margin, indicating our method is not sensitive to hyper-parameters.

5.2. Visualization Analysis

Visualization on Importance Score  In the reweighted learning strategy, the importance scores of each voxel or point are utilized to determine whether it should be distilled. Visualization of the importance scores in PointPillars is shown in Figure 6. It is observed that they successfully localize the foreground objects (e.g., cars and pedestrians) and the hard-negative objects (e.g., walls), indicating that the importance score in our method is able to find the voxels or points which are relatively more important.

Visualization on Detection Results  In this subsection, we have visualized the detection results of the student model trained with and without our method for comparison. Note that both student models are $4 \times$ compressed PointPillars trained on KITTI. The green and blue boxes indicate the boxes of the model prediction and the ground truth. As shown in Figure 7, the student model without knowledge distillation tends to have much more false-positive (FP) predictions. In contrast, this excessive FP problem is alleviated in the student trained with our method. This observation is consistent with our experimental results that the distilled PointPillars has 3.4 mAP improvements.

6. Conclusion

This paper proposes a structured knowledge distillation framework named PointDistiller for point clouds-based object detection. It is composed of local distillation to first encode the semantic information in local geometric structure in point clouds and distill it to students, and reweighted learning to handle the sparsity and noise in point clouds by assigning different learning weights to different points and voxels. Extensive experiments on both voxels-based detectors and raw points-based detectors have demonstrated the superiority over seven previous KD methods. Our ablation study has shown the individual effectiveness of each module in PointDistiller. Besides, the visualization results demonstrate that PointDistiller can significantly improve detection performance by reducing false-positive predictions, and the importance score is able to reveal the more significant voxels. To the best of our knowledge, this work initiates the first step to exploring KD for efficient point clouds-based 3D object detection, and we hope this could spur future research.
References


