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

Linfeng Zhang [1†] Runpei Dong [2†] Hung-Shuo Tai [3] Kaisheng Ma [1]
Tsinghua University [1] Xi'an Jiaotong University [2] DIDI [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 [14] and the achievements in end-to-end 3D representation learning [46, 47] have boosted the developments of point cloud based segmentation, generation, and detection [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 [3, 6, 14] and virtual reality [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 [18,35,37,73], quantization [8,12,40], lightweight model design [21,38,51], and knowledge distillation [20].

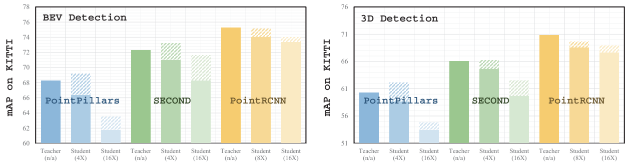


Figure 1. Experimental results (mAP of moderate difficulty) of our methods on 4×, 8×, and 16× compressed students on KITTI. The area of dash lines indicates the benefits of knowledge distillation.

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 [20, 50, 52, 66]. Sufficient theoretical and empirical results have demonstrated its effectiveness in image-based visual tasks such as image classification [20, 50], semantic segmentation [33] and object detection [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 [26, 39, 65]. (ii) Different from images that have a regular structure, point clouds are irregularly and sparsely distributed in the metric space [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

[†] The first two authors contribute equally. This work is done during the internship of L. Zhang in DIDI. K. Ma is the corresponding author.

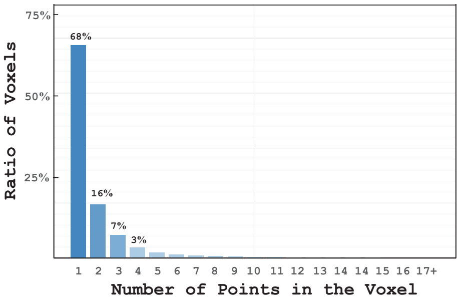


Figure 2. Distribution of the voxels with different number of points inside them. Voxels with no points are not included here.

detection [17, 53]. However, most of these methods focus on the choice of student-teacher in a multi-modal setting, *e.g.*, teaching point clouds-based student detectors with an images-based teacher or vice versa, and still ignore the peculiar properties of point clouds. To address this problem, we propose a structured knowledge distillation framework named PointDistiller, which involves *local distillation* to distill teacher knowledge in the local geometric structure of point clouds, and *reweighted learning* strategy to handle the sparsity of point clouds by highlight student learning on the relatively more crucial voxels or points.

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 Buciluǎ *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].

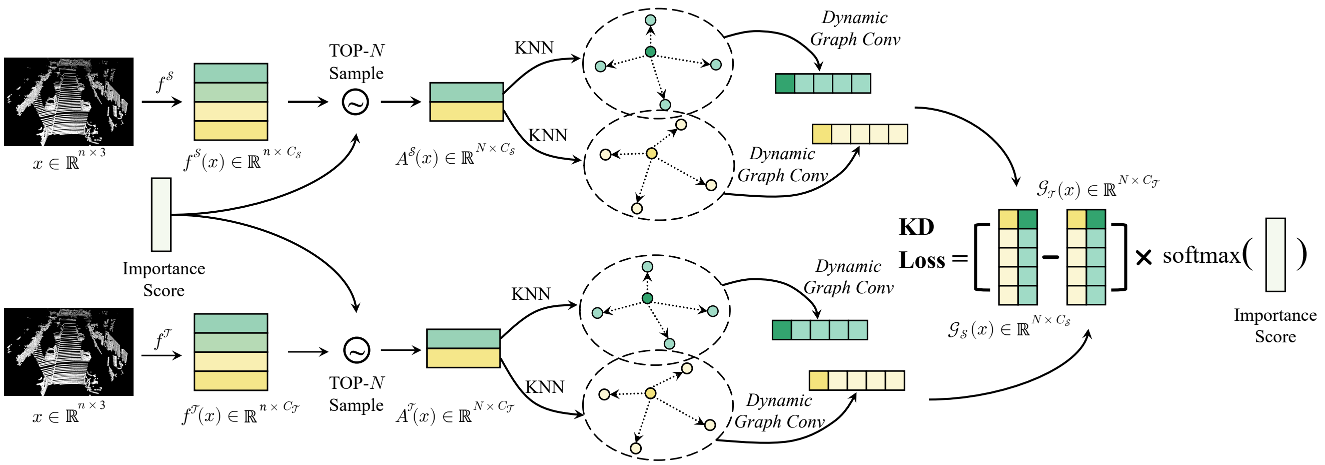
3. Methodology

3.1. Preliminaries

Given a set of point clouds $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ and the corresponding label set $\mathcal{Y} = \{y_1, y_2, ..., y_m\}$, the object detector can be formulated as $\mathcal{F} = f \circ g$, 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 $\mathcal{S}$ and $\mathcal{T}$, respectively.

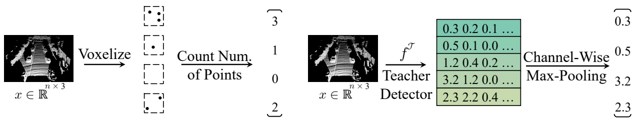
3.2. Our Method

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



$f^{\mathcal{S}}$
$x \in \mathbb{R}^{n \times 3}$
$f^{\mathcal{S}}(x) \in \mathbb{R}^{n \times C_{\mathcal{S}}}$
TOP-N Sample
$A^{\mathcal{S}}(x) \in \mathbb{R}^{N \times C_{\mathcal{S}}}$
KNN
KNN
Dynamic Graph Conv
Dynamic Graph Conv
Importance Score
$f^{\mathcal{T}}$
$x \in \mathbb{R}^{n \times 3}$
$f^{\mathcal{T}}(x) \in \mathbb{R}^{n \times C_{\mathcal{T}}}$
TOP-N Sample
$A^{\mathcal{T}}(x) \in \mathbb{R}^{N \times C_{\mathcal{T}}}$
KNN
KNN
Dynamic Graph Conv
Dynamic Graph Conv
$\mathcal{G}_{\mathcal{T}}(x) \in \mathbb{R}^{N \times C_{\mathcal{T}}}$
KD Loss =
× softmax( )
$\mathcal{G}_{\mathcal{S}}(x) \in \mathbb{R}^{N \times C_{\mathcal{S}}}$
Importance Score

Figure 3. The overview of our method. $f^{\mathcal{T}}$ and $f^{\mathcal{S}}$: the feature encoding layers in the teacher and student detectors. $A^{\mathcal{T}}$ and $A^{\mathcal{S}}$: features of the sampled to-be-distilled voxels or points with top-N largest importance score. $C_{\mathcal{T}}$ and $C_{\mathcal{S}}$: the number of channels for features of the teacher and the student detectors. $\mathcal{G}_{\mathcal{T}}$ and $\mathcal{G}_{\mathcal{S}}$: 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.



(a) *for voxels-based detectors*

(b) *for raw points-based detectors*

Figure 4. The computation of the importance score for voxels-based and raw-points-based detectors. The importance scores are later utilized to determine which voxel or point is utilized for distillation and how they contribute to the distillation loss.

on the importance score, we can sample the top-N significant voxels or points for knowledge distillation based on the importance score computed from $f^{\mathcal{T}}(x)$. For simplicity in writing, we denote the selected student and teacher features in top-N important voxels or points as $A^{\mathcal{T}}(x) \in \mathbb{R}^{N \times C_{\mathcal{T}}}$ and $A^{\mathcal{S}}(x) \in \mathbb{R}^{N \times C_{\mathcal{S}}}$, respectively, where $C_{\mathcal{S}}$ and $C_{\mathcal{T}}$ indicate the number of channels in student and teacher features.

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 $\mathcal{N}_i = \{z_{i,2}, z_{i,3}, ..., z_{i,K}\}$ 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 $\hat{z}_{i,j} = \text{cat}\big([z_{i,1}, z_{i,j}]\big)$ for all $z_{i,j} \in \mathcal{N}_i$. Then, we apply a dynamic graph convolution as the aggregation operation upon them, which can be formulated as $\mathcal{G}_i = \gamma(\hat{z}_{i,1}, ..., \hat{z}_{i,K})$, where γ is the aggregation operator. Following previous graph-based point cloud networks, we set γ as a nonlinear layer with ReLU activation and batch normalization. Then the training objective of local distillation can be formulated as

$$\underset{\theta_{\mathcal{S}}, \theta_{\gamma}}{\arg\min}\, \mathbb{E}_{\,x} \left[\frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{G}_i^{\mathcal{S}}(x) - \mathcal{G}_i^{\mathcal{T}}(x) \right\| \right], \tag{1}$$

where $\theta_{\mathcal{S}}$ indicates the parameters of student encoding layer $f^{\mathcal{S}}$. $\theta_{\gamma} = [\theta_{\gamma^{\mathcal{S}}}, \theta_{\gamma^{\mathcal{T}}}]$ 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 $\phi \in \mathbb{R}^N$.

Model	F	P	KD	Car			Pedestrians			Cyclists			mAP
				Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
PointPillars	34.3	4.8	×	94.3	88.1	83.6	57.9	51.8	47.6	86.5	65.0	61.1	68.3
	9.0	1.3	×	92.4	88.2	83.6	53.0	47.9	44.1	81.8	63.1	59.0	66.4
	9.0	1.3	✓	**93.1**	**89.0**	**86.3**	**59.8**	**52.8**	**48.2**	**83.8**	**65.8**	**62.0**	**69.2**
	2.5	0.3	×	91.3	84.8	**82.2**	50.1	44.4	41.6	74.2	56.1	52.5	61.8
	2.5	0.3	✓	**92.5**	**85.2**	81.9	**50.8**	**45.8**	**42.5**	**77.2**	**59.5**	**55.6**	**63.5**
SECOND	69.8	5.3	×	93.1	88.9	85.9	64.9	58.1	51.9	84.3	69.9	65.7	72.3
	17.8	1.4	×	93.1	86.6	85.7	64.7	57.8	52.8	84.1	68.5	64.5	71.0
	17.8	1.4	✓	**93.2**	**88.6**	**86.0**	**65.1**	**58.1**	**53.1**	**87.4**	**72.9**	**68.5**	**73.2**
	4.6	0.4	×	95.0	86.2	83.3	61.6	54.9	49.2	80.9	63.6	59.6	68.3
	4.6	0.4	✓	**95.4**	**88.3**	**83.7**	**64.5**	**57.6**	**52.2**	**85.2**	**68.8**	**64.4**	**71.6**
PointRCNN	104.9	4.1	×	95.0	86.7	84.3	69.8	64.5	58.1	92.8	74.6	70.4	75.3
	13.7	0.5	×	**93.5**	**85.9**	**83.5**	71.6	65.4	59.1	91.1	71.0	67.2	74.1
	13.7	0.5	✓	93.3	85.7	**83.5**	**74.0**	**67.2**	**60.5**	**94.6**	**72.3**	**67.9**	**75.1**
	7.1	0.3	×	**95.8**	**85.4**	**81.7**	**72.9**	**65.5**	**58.6**	91.8	69.3	65.9	73.4
	7.1	0.3	✓	95.2	84.3	**81.7**	72.6	64.8	57.7	**92.6**	**72.9**	**68.5**	**74.0**

Table 1. Experimental results of our method for BEV (Bird-Eye-View) object detection. **F** and **P** indicate the number of float operations (/G) and parameters (/M) of the detector, respectively. **mAP** indicates the mean average precision of moderate difficulty. **KD** indicates whether our method is utilized. The reported result in the first line of each detector is the performance of the teacher detector.

Model	F	P	KD	Car			Pedestrians			Cyclists			mAP
				Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
PointPillars	34.3	4.8	×	87.3	75.9	71.1	52.0	45.9	41.4	78.6	59.2	55.8	60.3
	9.0	1.3	×	87.4	75.9	71.0	48.2	43.0	38.7	74.1	57.2	53.3	58.7
	9.0	1.3	✓	**88.1**	**76.9**	**73.8**	**54.6**	**47.5**	**42.3**	**80.3**	**62.0**	**58.8**	**62.1**
	2.5	0.3	×	83.1	**69.8**	**65.4**	44.0	38.7	35.3	70.9	52.1	48.7	53.5
	2.5	0.3	✓	**83.7**	**69.8**	65.3	**45.3**	**40.3**	**36.5**	**72.7**	**54.7**	**51.1**	**54.9**
SECOND	69.8	5.3	×	88.6	79.3	75.7	60.1	53.2	47.0	79.8	65.7	61.6	66.1
	17.8	1.4	×	**89.2**	**77.4**	**74.0**	58.8	51.3	45.5	80.5	65.4	61.3	64.7
	17.8	1.4	✓	88.9	76.9	73.6	**60.0**	**53.0**	**47.4**	**83.2**	**68.6**	**64.2**	**66.2**
	4.6	0.4	×	86.3	72.6	66.0	53.6	47.8	41.8	76.7	58.7	55.1	59.7
	4.6	0.4	✓	**87.0**	**73.3**	**68.1**	**57.0**	**51.0**	**45.4**	**81.0**	**63.5**	**59.3**	**62.6**
PointRCNN	104.9	4.1	×	92.1	80.1	77.4	66.8	60.3	54.3	92.1	72.3	67.8	70.9
	13.7	0.5	×	89.8	**76.8**	72.7	67.9	60.9	54.0	88.1	68.0	64.4	68.6
	13.7	0.5	✓	**91.4**	75.6	**72.9**	**70.1**	**63.5**	**56.1**	**92.0**	**69.8**	**65.4**	**69.6**
	7.1	0.3	×	**89.8**	75.3	70.7	68.7	60.7	53.4	**91.1**	67.2	63.9	67.7
	7.1	0.3	✓	89.6	**75.6**	**72.6**	**69.4**	**61.0**	**53.5**	91.0	**70.2**	**65.5**	**69.0**

Table 2. Experimental results of our method for 3D object detection. **F** and **P** indicate the number of float operations (/G) and parameters (/M) of the detector, respectively. **mAP** indicates the mean average precision of moderate difficulty. **KD** indicates whether our method is utilized. The reported result in the first line of each detector is the performance of the teacher detector.

Similar with the importance score defined during sampling, we define the learning weight of each graph as the maximal value on the corresponding features after a softmax function, which can be formulated as $\phi = \text{softmax}\left(\max(G^{\mathcal{T}}(x))/\tau\right)$, where τ is the temperature hyper-parameter in softmax function. For voxels-based methods, we define ϕ as the number of points in the voxel after a softmax function, which can be formulated as $\phi_i = \text{softmax}\left(\sum_j v_{i,j}/\tau\right)$. Then, with the reweighting strategy, the training objective of knowledge distillation can be formulated as

$$\arg\min_{\theta_{\mathcal{S}},\theta_{\gamma}} \mathbb{E}_{\,x} \left[\frac{1}{N} \sum_{i=1}^{N} \phi_i \cdot \left\| \mathcal{G}_i^{\mathcal{S}}(x) - \mathcal{G}_i^{\mathcal{T}}(x) \right\| \right]. \quad (2)$$

Task	Method	Car			Pedestrians			Cyclists			mAP
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
BEV	Teacher w/o KD	94.3	88.1	83.6	57.9	51.8	47.6	86.5	65.0	61.1	68.3
	Student w/o KD	92.4	88.2	83.6	53.0	47.9	44.1	81.8	63.1	59.0	66.4
	+ Romero *et al.* [50]	91.5	85.6	83.1	57.5	51.0	46.3	82.8	65.1	61.1	67.2
	+ Zagoruyko *et al.* [69]	92.6	88.0	83.6	56.7	50.9	47.3	81.4	64.4	60.5	67.7
	+ Zheng *et al.* [75]	92.7	87.9	83.2	57.7	51.0	46.8	78.1	61.8	57.9	66.9
	+ Tung *et al.* [60]	92.8	88.0	83.3	54.5	48.7	45.2	84.2	64.3	60.7	67.0
	+ Tian *et al.* [59]	92.7	87.8	83.2	56.6	50.4	46.8	80.3	61.9	57.9	66.7
	+ Heo *et al.* [19]	92.6	87.9	83.5	57.6	51.0	46.8	78.1	61.8	57.8	66.9
	+ Zhang *et al.* [71]	92.3	85.7	83.0	59.7	52.0	47.6	71.0	64.3	60.5	67.5
	+ Ours	**93.1**	**89.0**	**86.3**	**59.8**	**52.8**	**48.2**	**83.8**	**65.8**	**62.0**	**69.2**
3D	Teacher w/o KD	87.3	75.9	71.1	52.0	45.9	41.4	78.6	59.2	55.8	60.3
	Student w/o KD	87.4	75.9	71.0	48.2	43.0	38.7	74.1	57.2	53.3	58.7
	+ Romero *et al.* [50]	84.9	73.4	70.6	50.9	44.2	39.3	75.9	58.5	54.6	58.7
	+ Zagoruyko *et al.* [69]	87.6	75.7	71.4	51.0	44.8	40.7	74.4	57.8	54.2	59.5
	+ Zheng *et al.* [75]	87.3	75.5	71.5	52.6	45.6	40.8	74.9	58.6	54.9	59.9
	+ Tung *et al.* [60]	87.5	76.0	71.3	50.1	43.3	39.2	79.2	59.5	55.3	59.6
	+ Tian *et al.* [59]	85.6	74.2	71.0	49.5	43.5	39.0	76.4	58.4	54.7	58.7
	+ Heo *et al.* [19]	87.7	76.1	71.7	52.6	45.6	40.8	74.9	58.6	54.9	60.1
	+ Zhang *et al.* [71]	87.5	75.8	71.6	53.4	45.8	40.9	76.1	59.0	55.2	60.2
	+ Ours	**88.1**	**76.9**	**73.8**	**54.6**	**47.5**	**42.3**	**80.3**	**62.0**	**58.8**	**62.1**

Table 3. Comparison between our method and previous knowledge distillation methods on PointPillars. The teacher and the student detectors have 34.3 and 9.0 GFLOPs, respectively. **mAP** indicates the mean average precision of moderate difficulty.

		BEV Detection										3D Detection									
LD	RL	Car			Pedestrians			Cyclists			mAP	Car			Pedestrians			Cyclists			mAP
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
×	×	92.4	88.2	83.6	53.0	47.9	44.1	81.8	63.1	59.0	66.4	87.4	75.9	71.0	48.2	43.0	38.7	74.1	57.2	53.3	58.7
✓	×	92.7	88.2	83.7	58.2	51.0	47.0	**84.3**	66.9	**63.1**	68.7	87.6	76.0	71.5	52.6	45.9	40.7	79.8	61.6	58.0	61.2
×	✓	**93.1**	88.5	85.7	55.6	49.6	45.7	84.2	**67.3**	62.9	68.4	87.8	76.5	72.0	49.4	43.7	39.4	78.7	61.5	57.5	60.6
✓	✓	**93.1**	**89.0**	**86.3**	**59.8**	**52.8**	**48.2**	83.8	65.8	62.0	**69.2**	**88.1**	**76.9**	**73.8**	**54.6**	**47.5**	**42.3**	**80.3**	**62.0**	**58.8**	**62.1**

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 the above loss function, with a higher ϕ_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

Model	FLOPs (/G)	#Params (/M)	KD Method	mAP(↑)	NDS(↑)
PointPillars	34.3	4.8	Teacher w/o KD	39.3	53.2
	17.1 (2.00×)	2.4 (2.00×)	Student w/o KD	36.0	50.5
			+ Heo *et al.* [19]	$36.2_{+0.2}$	$50.6_{+0.1}$
			+ Tian *et al.* [59]	$35.7_{-0.3}$	$50.4_{-0.1}$
			+ Wang *et al.* [62]	$36.2_{+0.2}$	$50.7_{+0.2}$
			+ Ours	$\mathbf{36.7}_{\mathbf{+0.7}}$	$\mathbf{51.0}_{\mathbf{+0.5}}$
	9.0 (3.8×)	1.3 (3.69×)	Student w/o KD	32.2	47.3
			+ Heo *et al.* [19]	$32.4_{+0.2}$	$47.6_{+0.3}$
			+ Tian *et al.* [59]	$32.3_{+0.1}$	$47.2_{+0.4}$
			+ Wang *et al.* [62]	$32.5_{+0.3}$	$47.8_{+0.5}$
			+ Ours	$\mathbf{32.8}_{\mathbf{+0.6}}$	$\mathbf{48.6}_{+1.3}$
CenterPoint	121.2	9.2	Teacher w/o KD	57.3	65.6
	45.6 (2.66×)	4.8 (1.92×)	Student w/o KD	55.4	64.2
			+ Heo *et al.* [19]	$55.7_{+0.3}$	$64.4_{+0.2}$
			+ Tian *et al.* [59]	$55.9_{+0.5}$	$64.6_{+0.4}$
			+ Wang *et al.* [62]	$55.8_{+0.4}$	$64.6_{+0.4}$
			+ Ours	$\mathbf{56.4}_{\mathbf{+1.0}}$	$\mathbf{65.1}_{\mathbf{+0.9}}$
	71.7 (1.69×)	6.3 (1.46×)	Student w/o KD	56.0	64.5
			+ Heo *et al.* [19]	$56.3_{+0.3}$	$64.8_{+0.3}$
			+ Tian *et al.* [59]	$56.5_{+0.5}$	$65.0_{+0.4}$
			+ Wang *et al.* [62]	$56.3_{+0.3}$	$64.9_{+0.4}$
			+ Ours	$\mathbf{57.0}_{\mathbf{+1.0}}$	$\mathbf{65.3}_{\mathbf{+0.8}}$

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

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 1.8 and 0.1 mAP, respectively. (iii) Consistent average precision boosts can be observed in detection results of all difficulties. For instance, on 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 on cars, pedestrians and cyclists, respectively. (v) On PointRCNN, on average 1.3 and 1.2 moderate mAP improvements can be observed on BEV and 3D detection, respectively, indicating that our method is also effective for raw points-based detectors. In summary, these experiment results demonstrate that our method can successfully transfer the knowledge from teacher detectors to student detectors and lead to significant and consistent performance boosts.

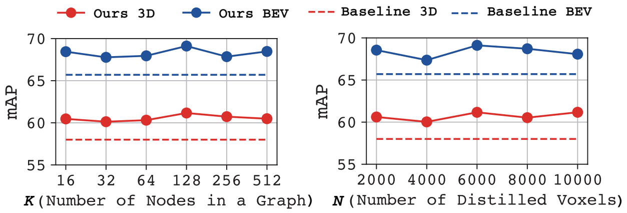


Figure 5. A sensitivity study to hyper-parameters in our method on KITTI with 4× compressed PointPillars detctors. mAP is measured on the moderate difficulty.

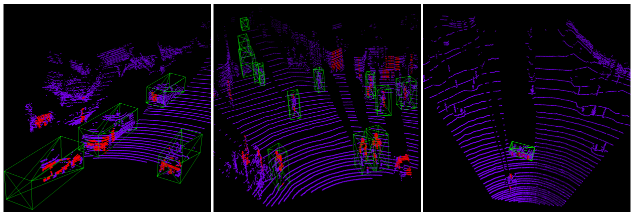


Figure 6. Visualization on importance scores for PointPillars. Red points indicate the voxels with high importance scores.

Student Latency We have also measured the latency of teachers and students on RTX 2080Ti. Our experiments show that the 4× compressed PointPillars and SECOND, the 8× compressed PointRCNN achieves 2.2×, 2.5×, 3.4× acceleration compared with their teachers in terms of latency, indicating the acceleration of our method is also effective on hardware. Please refer to our appendix for more details.

Comparison with Other KD Methods Comparison between our method and previous knowledge distillation methods is shown in Table 3. It is observed that: (i) Our method outperforms the previous methods by a clear margin. On BEV and 3D detection, our method outperforms the second-best KD method by 1.5 and 1.9 moderate mAP, respectively. (ii) Our method achieves the best performance for all categories of all difficulties. (iii) Besides, our method is the only knowledge distillation method that enables the student detector to outperform its teacher detector.

Experiments on nuScenes Experiments of around 2× and 4× compressed PointPillars and CenterPoint on nuScenes are shown in Table 5. It is observed that our method leads to 0.83 and 0.88 improvements on mAP and NDS on average, respectively, outperforming the other KD methods by a large margin. These observations indicate that our method is also effective on the large-scale dataset.

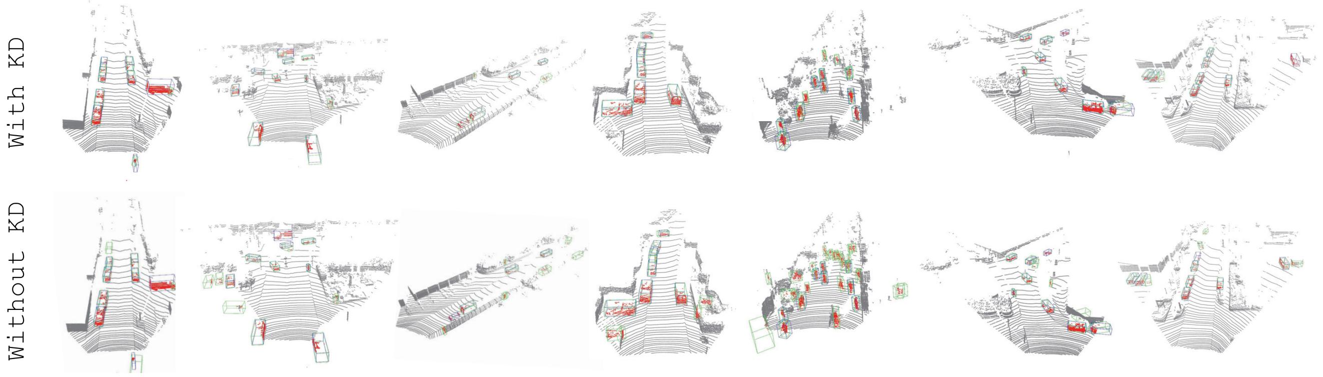


Figure 7. Comparison between the detection results of students trained with and without KD. Greens and blue boxes indicate the bounding boxes from prediction and ground-truths, respectively. Red points are the points insides the predicted bounding boxes.

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× 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× 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

[1] Mohammad Farhadi Bajestani and Yezhou Yang. Tkd: Temporal knowledge distillation for active perception. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pages 953–962, 2020. 1

[2] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, pages 535–541. ACM, 2006. 2

[3] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020. 1, 6

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Eur. Conf. Comput. Vis. (ECCV)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. 3

[5] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Adv. Neural Inform. Process. Syst. (NIPS)*, pages 742–751, 2017. 1, 2

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6526–6534. IEEE Computer Society, 2017. 1

[7] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Int. Conf. Comput. Vis. (ICCV)*, pages 9775–9784, 2019. 3

[8] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: parameterized clipping activation for quantized neural networks. *CoRR*, abs/1805.06085, 2018. 1

[9] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. Monodistill: Learning spatial features for monocular 3d object detection. In *Int. Conf. Learn. Represent. (ICLR)*, 2022. 3

[10] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020. 6

[11] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 7842–7851, 2021. 3

[12] Runpei Dong, Zhanhong Tan, Mengdi Wu, Linfeng Zhang, and Kaisheng Ma. Finding the task-optimal low-bit sub-distribution in deep neural networks. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 5343–5359. PMLR, 2022. 1

[13] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 1

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 6, 7

[15] Ben Graham. Sparse 3d convolutional neural networks. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Brit. Mach. Vis. Conf. (BMVC)*, pages 150.1–150.9. BMVA Press, 2015. 1

[16] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2154–2164. Computer Vision Foundation / IEEE, 2021. 3

[17] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Int. Conf. Comput. Vis. (ICCV)*, pages 3133–3143. IEEE, 2021. 2, 3

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Int. Conf. Learn. Represent. (ICLR)*, 2016. 1

[19] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Int. Conf. Comput. Vis. (ICCV)*, pages 1921–1930, 2019. 6, 7

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2014. 1, 2

[21] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *Int. Conf. Comput. Vis. (ICCV)*, pages 1314–1324. IEEE, 2019. 1

[22] Qing Jin, Jian Ren, Oliver J Woodford, Jiazhuo Wang, Geng Yuan, Yanzhi Wang, and Sergey Tulyakov. Teachers do more than teach: Compressing image-to-image models. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 13600–13611, 2021. 2

[23] Zijian Kang, Peizhen Zhang, Xiangyu Zhang, Jian Sun, and Nanning Zheng. Instance-conditional knowledge distillation for object detection. *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 34, 2021. 3

[24] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2017. 2

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 12697–12705, 2019. 1, 3, 6

[26] Guohao Li, Matthias Mueller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, pages 1–1, 2021. 1

[27] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional gans. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 5284–5294, 2020. 2

[28] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6356–6364, 2017. 1, 2

[29] Shaojie Li, Jie Wu, Xuefeng Xiao, Fei Chao, Xudong Mao, and Rongrong Ji. Revisiting discriminator in GAN compression: A generator-discriminator cooperative compression scheme. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, pages 28560–28572, 2021. 2

[30] Yawei Li, He Chen, Zhaopeng Cui, Radu Timofte, Marc Pollefeys, Gregory S Chirikjian, and Luc Van Gool. Towards efficient graph convolutional networks for point cloud handling. In *Int. Conf. Comput. Vis. (ICCV)*, pages 3752–3762, 2021. 3

[31] Zeqi Li, Ruowei Jiang, and Parham Aarabi. Semantic relation preserving knowledge distillation for image-to-image translation. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 648–663. Springer, 2020. 2

[32] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 7546–7555, 2021. 3

[33] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2604–2613, 2019. 1, 2

[34] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense prediction. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2020. 3

[35] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Int. Conf. Comput. Vis. (ICCV)*, October 2019. 1

[36] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, volume 32, 2019. 3

[37] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2018. 1

[38] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 116–131, 2018. 1

[39] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. 1

[40] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Int. Conf. Comput. Vis. (ICCV)*, October 2019. 1

[41] Anshul Paigwar, David Sierra-Gonzalez, Özgür Erkent, and Christian Laugier. Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar. In *Int. Conf. Comput. Vis. (ICCV)*, pages 2926–2933, 2021. 3

[42] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3967–3976, 2019. 2

[43] Youngmin Park, Vincent Lepetit, and Woontack Woo. Multiple 3d object tracking for augmented reality. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008*, pages 117–120. IEEE Computer Society, 2008. 1

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, pages 8024–8035, 2019. 6

[45] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Int. Conf. Comput. Vis. (ICCV)*, pages 5007–5016, 2019. 2

[46] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 652–660, 2017. 1, 3, 4

[47] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst. (NIPS)*, volume 30, 2017. 1, 2, 3, 4

[48] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6134–6144, 2021. 1, 3

[49] Yuxi Ren, Jie Wu, Xuefeng Xiao, and Jianchao Yang. Online multi-granularity distillation for GAN compression. In *Int. Conf. Comput. Vis. (ICCV)*, pages 6773–6783. IEEE, 2021. 2

[50] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Int. Conf. Learn. Represent. (ICLR)*, 2015. 1, 2, 6

[51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4510–4520, 2018. 1

[52] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 1, 2

[53] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 2, 3

[54] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10526–10535. Computer Vision Foundation / IEEE, 2020. 3

[55] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 770–779, 2019. 3, 6

[56] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Int. Conf. Comput. Vis. (ICCV)*, pages 3859–3868, 2019. 3

[57] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 685–702. Springer, 2020. 3

[58] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Adv. Neural Inform. Process. Syst. (NIPS)*, pages 1195–1204, 2017. 2

[59] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2020. 6, 7

[60] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Int. Conf. Comput. Vis. (ICCV)*, pages 1365–1374, 2019. 2, 6

[61] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4933–4942, 2019. 2, 3

[62] Yue Wang, Alireza Fathi, Jiajun Wu, Thomas Funkhouser, and Justin Solomon. Multi-frame to single-frame: knowledge distillation for 3d object detection. *arXiv preprint arXiv:2009.11859*, 2020. 7

[63] Yue Wang and Justin M. Solomon. Object DGCNN: 3d object detection using dynamic graphs. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, pages 20745–20758, 2021. 3

[64] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5):1–12, 2019. 2, 3, 4

[65] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9621–9630. Computer Vision Foundation / IEEE, 2019. 1

[66] Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. Bert-of-theseus: Compressing BERT by progressive module replacing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7859–7869. Association for Computational Linguistics, 2020. 1, 2

[67] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 3, 6

[68] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 6

[69] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Int. Conf. Learn. Represent. (ICLR)*, 2017. 2, 6

[70] Linfeng Zhang, Xin Chen, Xiaobing Tu, Pengfei Wan, Ning Xu, and Kaisheng Ma. Wavelet knowledge distillation: Towards efficient image-to-image translation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 2

[71] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. 1, 2, 3, 6

[72] Linfeng Zhang, Yukang Shi, Zuoqiang Shi, Kaisheng Ma, and Chenglong Bao. Task-oriented feature distillation. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2020. 2

[73] Tianyun Zhang, Shaokai Ye, Kaiqi Zhang, Jian Tang, Wujie Wen, Makan Fardad, and Yanzhi Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Eur. Conf. Comput. Vis. (ECCV)*, September 2018. 1

[74] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4320–4328, 2018. 2

[75] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. SE-SSD: self-ensembling single-stage object detector from point cloud. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 14494–14503. Computer Vision Foundation / IEEE, 2021. 6

[76] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Int. Conf. Comput. Vis. (ICCV)*, pages 4965–4974, 2021. 3

[77] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4490–4499, 2018. 3