itKD: Interchange Transfer-based Knowledge Distillation for 3D Object Detection

Hyeon Cho[1], Junyong Choi[1,2], Geonwoo Baek[1], and Wonjun Hwang[1,3]
[1]Ajou University, [2]Hyundai Motor Company, [3]Naver AI Lab
ch0104@ajou.ac.kr, chldusxkr@hyundai.com, bkw0622@ajou.ac.kr, wjhwang@ajou.ac.kr

Abstract

Point-cloud based 3D object detectors recently have achieved remarkable progress. However, most studies are limited to the development of network architectures for improving only their accuracy without consideration of the computational efficiency. In this paper, we first propose an autoencoder-style framework comprising channel-wise compression and decompression via interchange transfer-based knowledge distillation. To learn the map-view feature of a teacher network, the features from teacher and student networks are independently passed through the shared autoencoder; here, we use a compressed representation loss that binds the channel-wised compression knowledge from both student and teacher networks as a kind of regularization. The decompressed features are transferred in opposite directions to reduce the gap in the interchange reconstructions. Lastly, we present an head attention loss to match the 3D object detection information drawn by the multi-head self-attention mechanism. Through extensive experiments, we verify that our method can train the lightweight model that is well-aligned with the 3D point cloud detection task and we demonstrate its superiority using the well-known public datasets; e.g., Waymo and nuScenes.[1]

1. Introduction

Convolutional neural network (CNN)-based 3D object detection methods using point clouds [13] [35] [36] [43] [49] have attracted wide attention based on their outstanding performance for self-driving cars. Recent CNN-based works have required more computational complexity to achieve higher precision under the various wild situation. Some studies [23] [36] [43] have proposed methods to improve the speed of 3D object detection through which the non-maximum suppression (NMS) or anchor procedures are removed but the network parameters are still large.

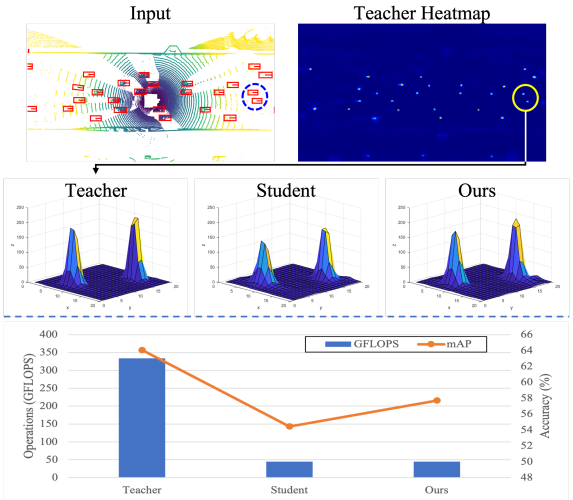


Figure 1. **Performance comparison between teacher and student networks for a point-cloud based 3D object detection.** The top example images are qualitatively compared between the results of teacher, student and our networks. Specifically, the first row images are an input sample with labels and the center heatmap head of the teacher network. The second row examples are responses of teacher, student, and ours for the yellow circle on the heatmap (or the blue dash circle on the input). The bottom image quantitatively shows the computational complexity and the corresponding accuracy of teacher, student and our networks, respectively. Best viewed in color.

Knowledge distillation (KD) is one of the parameter compression techniques, which can effectively train a compact student network through the guidance of a deep teacher network, as shown in the example images of Fig. 1. Starting with Hinton's work [9], many KD studies [10] [20] [28] [44] have transferred the discriminative teacher knowledge to the student network for classification tasks. From the viewpoint of the detection task, KD should be extended to the regression problem, including the object locations, which is not easy to straight-forwardly apply the classification-based KD methods to the detection task. To alleviate this problem, KD methods for object detection have been developed for mimicking the output of the backbone network [15] (*e.g.*, region

[1] Our code is available at https://github.com/hyeon-jo/interchange-transfer-KD.

proposal network) or individual detection head [2] [32]. Nevertheless, these methods have only been studied for detecting 2D image-based objects, and there is a limit to applying them to sparse 3D point cloud-based data that have not object-specific color information but only 3D position-based object structure information.

Taking a closer look at differences between 2D and 3D data, there is a large gap in that 2D object detection usually predicts 2D object locations based on inherent color information with the corresponding appearances, but 3D object detection estimates 3D object boxes from inputs consisting of only 3D point clouds. Moreover, the number of the point clouds constituting objects varies depending on the distances and presence of occlusions [42]. Another challenge in 3D object detection for KD is that, compared to 2D object detection, 3D object detection methods [4] [6] [43] [21] have more detection head components such as 3D boxes, and orientations. These detection heads are highly correlated with each other and represent different 3D characteristics. In this respect, when transferring the detection heads of the teacher network to the student network using KD, it is required to guide the distilled knowledge under the consideration of the correlation among the multiple detection head components.

In this paper, we propose a novel interchange transfer-based KD (itKD) method designed for the lightweight point-cloud based 3D object detection. The proposed itKD comprises two modules: (1) a channel-wise autoencoder based on the interchange transfer of reconstructed knowledge and (2) a head relation-aware self-attention on multiple 3D detection heads. First of all, through a channel-wise compressing and decompressing processes for KD, the interchange transfer-based autoencoder effectively represents the map-view features from the viewpoint of 3D representation centric-knowledge. Specifically, the encoder provides an efficient representation by compressing the map-view feature in the channel direction to preserve the spatial positions of the objects and the learning of the student network could be regularized by the distilled position information of objects in the teacher network. For transferring the interchange knowledge to the opposite networks, the decoder of the student network reconstructs the map-view feature under the guidance of the teacher network while the reconstruction of the teacher network is guided by the map-view feature of the student network. As a result, the student network can effectively learn how to represent the 3D map-view feature of the teacher. Furthermore, to refine the teacher's object detection results as well as its representation, our proposed head relation-aware self-attention gives a chance to learn the pivotal information that should be taught to the student network for improving the 3D detection results by considering the inter-head relation among the multiple detection head and the intra-head relation of the individual detection head.

In this way, we implement a unified KD framework to successfully learn the 3D representation and 3D detection results of the teacher network for the lightweight 3D point cloud object detection. We also conduct extensive ablation studies for thoroughly validating our approach in Waymo and nuScenes datasets. The results reveal the outstanding potential of our approach for transferring distilled knowledge that can be utilized to improve the performance of 3D point cloud object detection models.

Our contributions are summarized as follows:

- For learning the 3D representation-centric knowledge from the teacher network, we propose the channel-wise autoencoder regularized in the compressed domain and the interchange knowledge transfer method wherein the reconstructed features are guided by the opposite networks.
- For detection head-centric knowledge of the teacher, we suggest the head relation-aware self-attention which can efficiently distill the detection properties under the consideration of the inter-head relation and intra-head relation of the multiple 3D detection heads.
- Our work is the best attempt to reduce the parameters of point cloud-based 3D object detection using KD. Additionally, we validate its superiority using two large datasets that reflect real-world driving conditions, e.g., Waymo and NuScenes.

2. Related Works

2.1. 3D Object Detection based on Point Cloud

During the last few years, encouraged by the success of CNNs, the development of object detectors using CNNs is developing rapidly. Recently, many 3D object detectors have been studied and they can be briefly categorized by how they extract representations from point clouds; *e.g.*, grid-based [35] [36] [49] [13] [43], point-based [18] [23] [17] [25] [39] and hybrid-based [3] [40] [8] [48] [22] methods. In detail, Vote3Deep [5] thoroughly exploited feature-centric voting to build CNNs for detecting objects in point clouds. In [29], they have studied on the task of amodal 3D object detection in RGB-D images, where a 3D region proposal network (RPN) to learn objectness from geometric shapes and the joint object recognition network to extract geometric features in 3D and color features in 2D. The 3D fully convolutional network [14] was straightforwardly applied to point cloud data for vehicle detection. In the early days, VoxelNet [49] has designed an end-to-end trainable detector based on learning-based voxelization using fully connected layers. In [35], they encoded the point cloud by VoxelNet and used the sparse convolution to achieve the fast detection. HVNet [41] fused the multi-scale voxel feature encoder at the point-wise level and projected into multiple pseudo-image feature maps for solving the various sizes of the feature map. In [26], they replaced the point cloud with a grid-based bird's-eye view (BEV) RGB-map and utilized YOLOv2 to detect the 3D objects. PIXOR [36] converted the point cloud to a 3D BEV map and carried out the real-time 3D object detection with an RPN-free single-stage based model.

Recently, PointPillars (PP)-based method [13] utilized the PointNet [19] to learn the representation of point clouds organized in vertical columns for achieving the fast 3D object detection. To boost both performance and speed over PP, a pillar-based method [33] that incorporated a cylindrical projection into multi-view feature learning was proposed. More recently, CenterPoint [43] was introduced as an anchor-free detector that predicted the center of an object using a PP or VoxelNet-based feature encoder. In this paper, we employ the backbone architecture using CenterPoint because it is simple, near real-time, and achieves good performance in the wild situation.

2.2. Knowledge Distillation

KD is one of the methods used for compressing deep neural networks and its fundamental key is to imitate the knowledge extracted from the teacher network, which has heavy parameters as well as good accuracy. Hinton et al. [9] performed a knowledge transfer using KL divergence; FitNet [20] proposed a method for teaching student networks by imitating intermediate layers. On the other hand, TAKD [16] and DGKD [28] used multiple teacher networks for transferring more knowledge to the student network in spite of large parameter gaps. Recently, some studies have been proposed using the layers shared between the teacher and the student networks for KD. Specifically, in [37], KD was performed through softmax regression as the student and teacher networks shared the same classifier. IEKD [10] proposed a method to split the student network into inheritance and exploration parts and mimic the compact teacher knowledge through a shared latent feature space via an autoencoder.

Beyond its use in classification, KD for detection should transfer the regression knowledge regarding the positions of the objects to the student network. For this purpose, a KD for 2D object detection [15] was first proposed using feature map mimic learning. In [2], they transferred the detection knowledge of the teacher network using hint learning for an RPN, weighted cross-entropy loss for classification, and bound regression loss for regression. Recently, Wang et al. [32] proposed a KD framework for detection by utilizing the cross-location discrepancy of feature responses through fine-grained feature imitation.

As far as we know, there are few KD studies [7] [47] [34] [38] on point cloud-based 3D object detection so far. However, looking at similar studies on 3D knowledge transfer, SE-SSD [47] presented a knowledge distillation-based self-ensembling method for exploiting soft and hard targets with constraints to jointly optimize the model without extra computational cost during inference time. Object-DGCNN [34] proposed a NMS-free 3D object detection via dynamic graphs and a set-to-set distillation. They used the set-to-set distillation method for improving the performance without the consideration of the model compression. Another latest study is SparseKD [38] which suggested a label KD method that distills a few pivotal positions determined by teacher classification response to enhance the logit KD method. On the other hand, in this paper, we are more interest in how to make the student network lighter, or lower computational complexity, by using the KD for 3D object detection.

3. Methodology

3.1. Background

The 3D point cloud object detection methods [13] [49] generally consists of three components; a point cloud encoder, a backbone network, and detection heads. In this paper, we employ CenterPoint [43] network as a backbone architecture. Since the parameter size of the backbone network[2] is the largest among components of the 3D object detector, we aim to construct the student network by reducing the channel sizes of the backbone network for efficient network. We design our method to teach the student 3D representation-centric knowledge and detection head-centric knowledge of the teacher network, respectively.

3.2. Interchange Transfer

We adopt an autoencoder framework to effectively transfer the meaningful distilled knowledge regarding 3D detection from the teacher to the student network. The traditional encoder-based KD methods [10] [11] have been limited to the classification task, which transfers only compressed categorical knowledge to the student network. However, from the viewpoint of the detection task, the main KD goal of this paper is transferring the distilled knowledge regarding not only categorical features but also object location-related features. Particularly, unlike 2D detectors, 3D object detectors should regress more location information such as object orientations, 3D box sizes, etc., and it results in increasing the importance of how to transfer the 3D location features to the student network successfully.

For this purpose, we transfer the backbone knowledge that contains 3D object representation from the teacher network to the student through the compressed and reconstructed knowledge domains. As shown in Fig. 2, we in-

[2] The total parameter size of the 3D detector is about 5.2M and the backbone size is approximately 4.8M, which is 92%. Further details are found in the supplementary material.

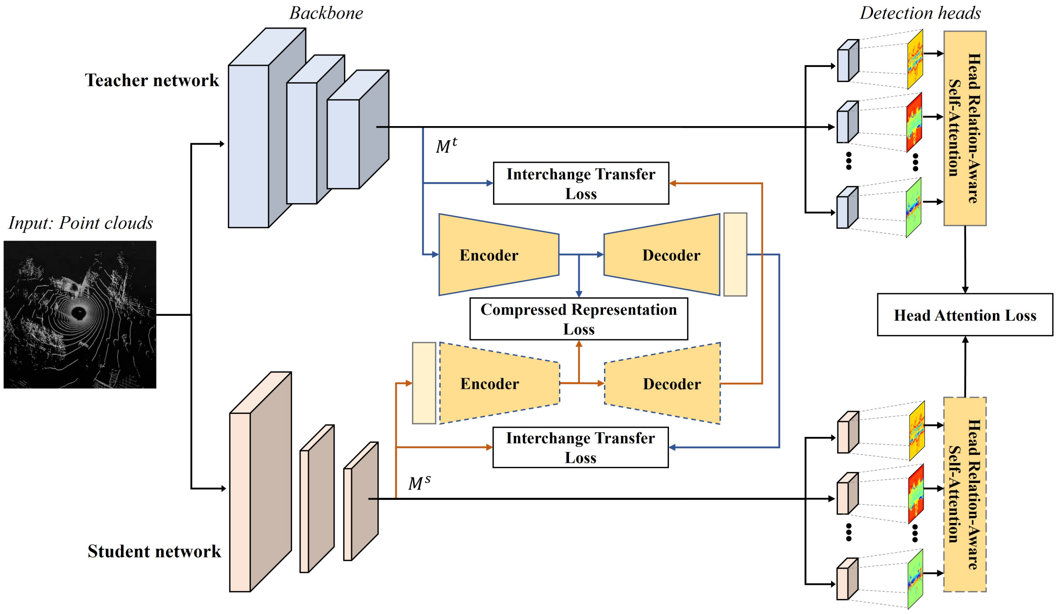

Backbone
Detection heads
Teacher network
Input: Point clouds
M^t
Interchange Transfer Loss
Encoder
Decoder
Compressed Representation Loss
Encoder
Decoder
Interchange Transfer Loss
M^s
Student network
Head Relation-Aware Self-Attention
Head Attention Loss
Head Relation-Aware Self-Attention

Figure 2. **Overview of the proposed knowledge distillation method.** The teacher and student networks take the same point clouds as inputs. Then, the map-view features M^t and M^s are extracted from the teacher and student networks, respectively. The channel-wise autoencoder transfers the knowledge obtained from M^t to M^s by using the compressed representation loss and interchange transfer loss consecutively. The head relation-aware self-attention provides the relation-aware knowledge of multiple detection head to the student network using the attention head loss. The dotted lines of the modules denote that there are shared network parameters between the teacher and student networks. The light-yellow boxes are buffer layers for sampling the features to match the channel sizes of networks.

troduce a channel-wise autoencoder which consists of an encoder in which the channel dimension of the autoencoder is gradually decreased and a decoder in the form of increasing the channel dimension. Note that spatial features play a pivotal role in the detection task and we try to preserve the spatial information by encoding features in the channel direction. We propose a compressed representation loss to coarsely guide location information of the objects to the student network in Fig. 2, and the compressed representation loss has an effect similar to the regularization of the autoencoder that binds the coordinates of the objectness between the teacher and student networks. The compressed representation loss function $\mathcal{L}_{cr}$ is represented as follows:

$$\begin{aligned} \mathcal{L}_{cr} &= m_{obj} \circ \mathcal{S}[E(\theta_{enc}, M^t), E(\theta_{enc}, M^s)] \\ &= m_{obj} \circ \mathcal{S}[M^t_{enc}, M^s_{enc}], \end{aligned} \tag{1}$$

where E is a shared encoder, which has the parameters θ_{enc}, and $\mathcal{S}$ denotes l_1 loss as a similarity measure. M^t and M^s are outputs of the teacher and student backbones, respectively. m_{obj} represents a binary mask to indicate object locations in backbone output like [38] and $\circ$ is an element-wise product.

After performing the coarse representation-based knowledge distillation in a compressed domain, the fine representation features of the teacher network are required to teach the student network from the viewpoint of 3D object detection. In this respect, the decoder reconstructs the fine map-view features in the channel direction from the compressed features. Through the proposed interchange transfer loss, the reconstructed features are guided from the opposite networks, not their own stem networks, as shown in Fig. 2. Specifically, since the teacher network is frozen and we use the shared autoencoder for both student and teacher networks, we can teach the reconstructed fine features from the student network to resemble the output of the teacher network M^t rather than the student M^s. Moreover, the reconstructed fine features from the teacher network can guide the student's output, M^s at the same time. The proposed interchange transfer loss $\mathcal{L}_{it}$ is defined as follows:

$$\mathcal{L}_{t \to s} = \mathcal{S}[M^s, D(\theta_{dec}, M^t_{enc})], \tag{2}$$

$$\mathcal{L}_{s \to t} = \mathcal{S}[M^t, D(\theta_{dec}, M^s_{enc})], \tag{3}$$

$$\mathcal{L}_{it} = \mathcal{L}_{s \to t} + \mathcal{L}_{t \to s}, \tag{4}$$

where D is the decoder that contains the network parameter θ_{dec}, which is a shared parameter. We hereby present the representation-based KD for 3D object detection in both compressed and decompressed domains to guide the student network to learn the map-view feature of the teacher network efficiently.

3.3. Head Relation-Aware Self-Attention

Fundamentally, our backbone network, *e.g.*, CenterPoint [43], has various types of 3D object characteristics

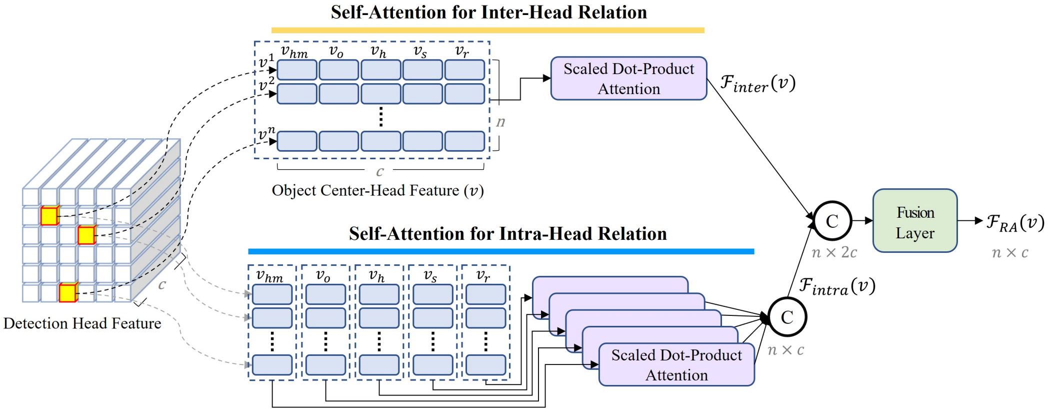

Self-Attention for Inter-Head Relation
v_{hm} v_o v_h v_s v_r
v^1
v^2
v^n
n
c
Object Center-Head Feature (v)
Scaled Dot-Product Attention
$\mathcal{F}_{inter}(v)$
C
$n \times 2c$
Fusion Layer
$\mathcal{F}_{RA}(v)$
$n \times c$
Detection Head Feature
c
Self-Attention for Intra-Head Relation
$\mathcal{F}_{intra}(v)$
C
$n \times c$
Scaled Dot-Product Attention

Figure 3. **Head Relation-Aware Self-Attention.** We make the object center-head feature from object center locations in the detection head feature and use it as different shaped inputs to self-attentions for inter-head relation and intra-head relation. In the self-attention for inter-head relation, we use the object center-head feature as an input for the self-attention. In the self-attention for intra-head relation, the detection heads are separately used for the independent self-attention functions. The outputs of the self-attentions are concatenated by ⓒ operations and the head relation-aware self-attention is generated through the fusion layer.

on detection heads. Specifically, the locations, size, and direction of an object are different properties, but they are inevitably correlated to each other because they come from the same object. However, the traditional KD methods [2] [34] were only concerned with how the student network straight-forwardly mimicked the outputs of the teacher network without considering the relation among the detection heads. To overcome this problem, we make use of the relation of detection heads as a major factor for the detection head-centric KD.

Our proposed head relation-aware self-attention is directly inspired by the multi-head self-attention [31] in order to learn the relation between the multiple detection head. As shown in Fig. 3, we first extract i-th instance feature $v^i \in \mathbb{R}^c$, where c is the channel size, from the center location of the object in the detection head feature. Note that, since the instance feature is extracted from the multiple detection head, it has several object properties such as a class-specific heatmap v^i_{hm}, a sub-voxel location refinement v^i_o, a height-above-ground v^i_h, a 3D size v^i_s, and a yaw rotation angle v^i_r. When there are a total of n objects, we combine them to make an object center-head feature $v \in \mathbb{R}^{n \times c}$. We use the same object center-head feature v of dimension n for query, key, and value, which are an input of the scaled dot-product attention. The self-attention function $\mathcal{F}$ is computed by

$$\mathcal{F}(v) = softmax(\frac{v^\top \cdot v}{\sqrt{n}}) \cdot v. \tag{5}$$

The proposed head relation-aware self-attention consists of two different self-attentions for inter-head and intra-head relations as illustrated in Fig. 3. We propose the self-attention based on the inter-head relation of the instance features, which is made in order to consider the relation between all detected objects and their different properties, rather than a single detected instance, from the global viewpoint. The self-attention for inter-head relation is computed by

$$\mathcal{F}_{inter}(v) = \mathcal{F}([v_{hm}, v_o, v_h, v_s, v_r]). \tag{6}$$

On the other hand, we suggest the self-attention for intra-head relation using the individual detection heads. Here we perform the attentions using only local relation in individual detection heads designed for different properties (e.g., orientation, size, etc.) and concatenate them. Its equation is

$$\mathcal{F}_{intra}(v) = [\mathcal{F}(v_{hm}), \mathcal{F}(v_o), \mathcal{F}(v_h), \mathcal{F}(v_s), \mathcal{F}(v_r)]. \tag{7}$$

We concatenate the outputs of the self-attentions and apply the fusion layer to calculate a final attention score that considers the relation between the detection heads and objects. The head relation-aware self-attention equation $\mathcal{F}_{RA}$ is derived by:

$$\mathcal{F}_{RA}(v) = \mathcal{G}([\mathcal{F}_{inter}(v), \mathcal{F}_{intra}(v)]), \tag{8}$$

where $\mathcal{G}$ is the fusion layer, e.g., 1×1 convolution layer. The student network indirectly takes the teacher's knowledge by learning the relation between the multiple detection head of the teacher network through head attention loss as follows:

$$\mathcal{L}_{attn} = \mathcal{S}(\mathcal{F}_{RA}(v_t), \mathcal{F}_{RA}(v_s)), \tag{9}$$

where v_t and v_s are the object center-head features of the teacher and the student, respectively.

Consequently, the overall loss is derived by

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{sup} + \beta(\mathcal{L}_{it} + \mathcal{L}_{cr} + \mathcal{L}_{attn}), \tag{10}$$

where $\mathcal{L}_{sup}$ is the supervised loss that consists of focal loss and regression loss, and α and β are the balancing parameters, which we set as 1 for simplicity.

4. Experimental Results and Discussions

4.1. Environment Settings

Waymo Waymo open dataset [30] is one of the large-scale datasets for autonomous driving, which is captured by the synchronized and calibrated high-quality LiDAR and camera across a range of urban and suburban geographies. This dataset provides 798 training scenes and 202 validation scenes obtained by detecting all the objects within a 75m radius; it has a total of 3 object categories (*e.g.*, vehicle, pedestrian, and cyclist) which have 6.1M, 2.8M, and 67K sets, respectively. The mean Average Precision (mAP) and mAP weighted by heading accuracy (mAPH) are the official metrics for Waymo evaluation. mAPH is a metric that gives more weight to the heading than it does to the sizes, and it accounts for the direction of the object.

nuScenes nuScenes dataset [1] is another large-scale dataset used for autonomous driving. This dataset contains 1,000 driving sequences. 700, 150, and 150 sequences are used for training, validation, and testing, respectively. Each sequence is captured approximately 20 seconds with 20 FPS using the 32-lane LiDAR. Its evaluation metrics are the average precision (AP) and nuScenes detection score (NDS). NDS is a weighted average of mAP and true positive metrics which measures the quality of the detections in terms of box location, size, orientation, attributes, and velocity.

Implementation details Following the pillar-based CenterPoint [43] as the teacher network, we use an Adam optimizer [12] with a weight decay of 0.01 and a cosine annealing strategy [27] to adjust the learning rate. We set 0.0003 for initial learning rate, 0.003 for max learning rate, and 0.95 for momentum. The networks have been trained for 36 epochs on 8×V100 GPUs with a batch size of 32. For Waymo dataset, we set the detection range to [-74.88m, 74.88m] for the X and Y axes, [-2m, 4m] for the Z-axis, and a grid size of (0.32m, 0.32m). In experiments on nuScenes dataset, we used a (0.2m, 0.2m) grid and set the detection range to [-51.2m, 51.2m] for the X and Y-axes, [-5m, 3m] for the Z-axis, and a grid size of (0.2m, 0.2m). Compared to the teacher network, the student network has $1/4$ less channel capacity of backbone network. Our channel-wise autoencoder consists of three 1×1 convolution layers as the encoder and three 1×1 convolution layers as the decoder and the number of filters are 128, 64, 32 in encoder layers and 64, 128, 384 in decoder layers. The student's input buffer layer increases the channel size of 196 to 384 and the teacher's output buffer layer decreases the channel size 384 to 196.

4.2. Overall KD Performance Comparison

We validate the performance of our method compared with well-known KD methods on the Waymo and nuScenes datasets. We re-implement the seven KD methods from 2D classification-based KD to 3D detection-based KD in this paper. We set the baseline by applying the Kullback-Leibler (KL) divergence loss [9] to the center heatmap head and l_1 loss to the other regression heads. FitNet [20] is a method that mimics the intermediate outputs of layers and we apply it to the output of the backbone for simplicity. We also simply extend EOD-KD [2], one of the 2D object detection KDs, to 3D object detection. We apply TOFD [45], a 3D classification-based KD, to our detection task and straight-forwardly use SE-SSD [47], Object DGCNN [34], and SparseKD [38] for 3D object detection KD.

Table 1 shows that our method almost outperforms other KD methods on mAP and mAPH values for level 1 and level 2 under all three categories of objects. Especially, our performance improvement of mAPH is better than other methods, which indicates our method guides the student network well where the detected objects are facing. To verify the generality of the proposed method, we make additional comparison results using the nuScenes dataset, another large-scale 3D dataset for autonomous driving, in Table 2. Compared with the other methods, our method achieves the best accuracy under the NDS and mAP metrics in the nuScenes validation set. Specifically, when the student network shows 50.24% NDS and 38.52% mAP, our method achieves 53.90% (+3.66%) NDS and 41.33% (+2.81%) mAP. In detail, our method outperforms the other methods for the most of object classes except the construction vehicle and the bicycle.

4.3. Ablation Studies

To analyze of our proposed method in detail, we conduct ablation studies on the Waymo dataset, and the whole performances are measured by mAPH at level 2 for simplicity. For the qualitative analysis, we visualize the map-view feature at each stage to validate the what kinds of knowledge are transferred from the teacher to the student by the proposed method. For simple visualization, we apply the L_1 normalization to the map-view feature in the channel direction.

As shown in Fig. 4, the objects and backgrounds are well activated in the example image of the teacher output. On the other hand, the encoder output is activated by further highlighting the coarse positions of the target objects. When looking at the decoder output, we can see that all the fine surrounding information is represented again. At this point, it is worth noting that compared to the teacher output, the target objects are highlighted a little more. From these visual comparisons, we can infer how our method successfully transfers the object-centered knowledge to the student.

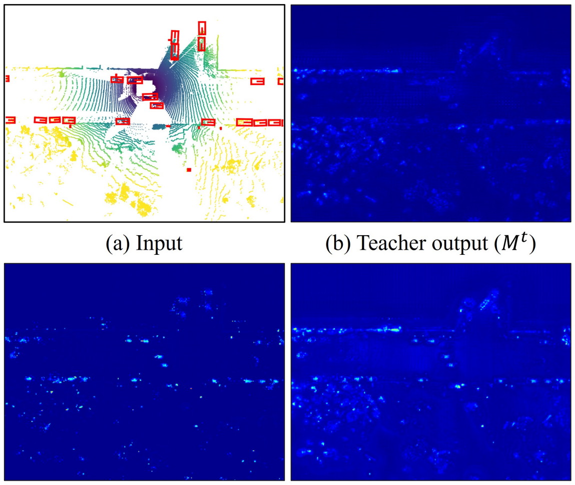
We explore the buffer layer that matches the channel size of the channel-wise autoencoder without the head attention loss. As shown in Table 3, we compare the three types for the buffer layer: (1) $S \rightarrow T$ is the upsampling method that in-

Table 1. **Waymo evaluation.** Comparisons with different KD methods in the Waymo validation set. The best accuracy is indicated in bold, and the second-best accuracy is underlined.

Method	Vehicle				Pedestrian				Cyclist			
	Level 1		Level 2		Level 1		Level 2		Level 1		Level 2	
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
Teacher [43]	73.72	73.17	65.61	65.11	72.43	61.72	64.73	54.99	64.30	62.61	61.91	60.28
Student (1/4)	64.22	63.56	56.21	55.62	63.72	53.22	56.14	46.78	53.01	51.72	50.99	49.75
Baseline	64.78	64.05	56.92	56.26	64.85	52.98	57.37	46.75	54.71	52.46	52.65	50.48
FitNet [20]	65.11	64.38	57.24	56.58	64.89	53.29	57.37	47.00	54.91	52.61	52.84	50.63
EOD-KD [2]	66.50	65.79	58.56	57.92	65.99	54.58	58.48	48.25	55.18	52.93	53.10	50.94
SE-SSD [47]	65.95	65.22	58.05	57.40	65.39	53.98	57.92	47.69	55.01	52.98	52.94	50.99
TOFD [45]	64.09	63.43	56.13	55.55	66.24	54.98	58.50	48.45	54.95	53.06	52.86	51.04
Obj. DGCNN [34]	66.07	65.38	59.27	58.55	65.98	54.44	59.42	49.11	54.65	52.62	53.13	50.93
SparseKD [38]	65.25	64.59	56.97	56.38	**67.44**	54.54	59.24	47.83	55.54	53.45	53.63	51.61
Ours	**67.43**	**66.72**	**59.44**	**58.81**	67.26	**56.02**	**59.73**	**49.61**	**56.09**	**54.24**	**53.96**	**52.19**

Table 2. **nuScenes evaluation.** Comparisons with different KD methods in the nuScenes validation set. The best accuracy is indicated in bold, and the second-best accuracy is underlined.

Method	NDS	mAP	car	truck	bus	trailer	con. veh.	ped.	motor.	bicycle	tr. cone	barrier
Teacher [43]	60.16	50.25	84.04	53.48	64.29	31.90	12.50	78.93	44.01	18.18	54.87	60.30
Student (1/4)	50.24	38.52	77.85	38.18	51.38	22.33	3.95	71.51	23.90	3.51	43.03	49.56
Baseline	51.48	39.19	78.72	37.90	50.47	22.42	3.51	72.29	26.25	4.65	44.91	50.77
FitNet [20]	51.42	38.90	78.30	37.40	50.40	22.20	3.80	72.10	25.70	4.25	44.20	50.60
EOD-KD [2]	52.49	39.82	78.40	38.60	50.90	22.70	3.90	73.20	28.20	5.30	45.00	51.97
SE-SSD [47]	52.21	39.53	78.69	38.56	49.81	23.70	3.72	72.86	28.27	4.25	44.24	51.18
TOFD [45]	52.88	40.57	79.06	39.73	52.03	24.51	3.56	73.51	29.58	5.62	45.34	52.79
Obj. DGCNN [34]	52.91	40.34	78.95	39.24	53.37	23.96	**4.13**	72.98	28.63	4.99	44.72	52.46
SparseKD [38]	53.01	40.26	78.78	39.50	51.87	23.64	3.30	73.17	29.34	**5.75**	44.98	52.26
Ours	**53.90**	**41.33**	**79.48**	**40.38**	**54.35**	**26.44**	3.58	**73.91**	**30.21**	5.39	**45.90**	**53.70**



(a) Input

(b) Teacher output (M^t)

(c) Encoder output

(d) Decoder output

Figure 4. **Feature visualization on the proposed channel-wise autoencoder.** (a) an example input image and (b) the output feature of the teacher network. (c) and (d) are the output images of encoder and decoder of the teacher, respectively.

Table 3. **Buffer layer for different channel size.**

Method	Vehicle	Pedestrian	Cyclist	Avg.
S → T	58.41	**48.90**	**51.90**	**53.07**
T → S	**58.62**	48.78	51.75	53.05
(S + T) / 2	58.47	48.84	51.54	52.95

Table 4. **Effect of shared and non-shared parameters for the autoencoder.**

Method	Vehicle	Pedestrian	Cyclist	Avg.
Non-shared	56.26	45.85	48.23	50.11
Shared	**58.41**	**48.90**	**51.90**	**53.07**

creases the student's map-view feature to the teacher's feature. (2) $T \rightarrow S$ is the downsampling method that decreases the teacher's feature to the student's feature. (3) *(S + T) / 2* is that the teacher's feature is downsampled and the student's feature is upsampled to the median size. The experiments show that the upsampling method performs better when considering all the classes.

In Table 4, we observe the performance difference when the autoencoder parameters are shared or not. From the result, we can conclude that the shared parameters achieve better performance because what we want to is for the student to learn the teacher's knowledge, not the independent model.

We investigate improvements made by our interchange transfer for KD without the head attention loss as shown in Table 5. Self-reconstruction is a method wherein the de-

Table 5. **Comparison of different reconstruction methods for the autoencoder.**

Method	Vehicle	Pedestrian	Cyclist	Avg.
Self Recon.	56.57	47.26	50.29	51.37
Ours	**58.41**	**48.90**	**51.90**	**53.07**

Table 6. **Comparison of KD methods for the multiple detection head.** KL loss and l_1 loss denote that directly apply the loss function to all detection heads for KD.

Method	Vehicle	Pedestrian	Cyclist	Avg.
Student	55.62	46.78	49.75	50.72
Baseline	56.26	46.75	50.48	51.16
KL loss [9]	55.92	45.08	47.49	49.50
l_1 loss	55.62	45.10	48.73	49.82
AT [44]	56.85	47.34	50.36	51.52
$\mathcal{L}_{inter}$	56.41	46.90	50.90	51.40
$\mathcal{L}_{intra}$	57.20	47.19	51.23	51.87
$\mathcal{L}_{attn}$	**57.10**	**47.34**	**51.79**	**52.08**

coder uses the corresponding input for the reconstruction and our interchange reconstruction is a method wherein the proposed $\mathcal{L}_{it}$ objective transfers the reconstructed knowledge to the opponent network. Our interchange transfer-based reconstruction achieves better results and note that our main task is not the reconstruction but the 3D object-based knowledge transfer for KD.

3D detection [4] [6] [43] [21] has the multiple detection head. To prove the superiority of the proposed head attention objective for 3D object detection, we make the KD comparison results against only multiple detection head without the autoencoder, as shown in Table 6. Since the heatmap head classifies objects and other heads regress 3D bounding box information, Applying KL loss and l_1 loss to all detection heads has a negative effect. However, it is required to consider the relation of detection heads. In this respect, our method achieves better performance than the other KD methods which directly mimic the output of detection heads or simply employ attention mechanism.

Table 7 shows the overall effect of the proposed losses on the KD performances. We set up the experiments by adding each loss based on the supervised loss $\mathcal{L}_{sup}$. Specifically, the interchange transfer loss $\mathcal{L}_{it}$ improves on an average of 1.41% mAPH and the compressed representation loss $\mathcal{L}_{cr}$ leads to a 0.94% performance improvement. In the end, the head attention loss $\mathcal{L}_{attn}$ helps to improve the performance and the final average mAPH is 53.54%. We conclude that each proposed loss contributes positively to performance improvement in the 3D object detection-based KD task.

Table 7. **Ablation results from investigating effects of different components.**

$\mathcal{L}_{sup}$	$\mathcal{L}_{it}$	$\mathcal{L}_{cr}$	$\mathcal{L}_{attn}$	Vehicle	Pedestrian	Cyclist	Avg.
✓				55.62	46.78	49.75	50.72
✓	✓			57.41	48.20	50.77	52.13
✓	✓	✓		58.41	48.90	51.90	53.07
✓	✓	✓	✓	**58.81**	**49.61**	**52.19**	**53.54**

Table 8. **Quantitative evaluation for model efficiency on Waymo dataset.**

Method	Params (M)	FLOPS (G)	mAPH / L2
PointPillars [13]	4.8	255.0	57.05
SECOND [35]	5.3	84.5	57.23
Part-A^2 [24]	4.6	87.1	57.43
IA-SSD [46]	2.7	46.1	58.08
SparseKD-v0.64 [38]	5.2	85.1	58.89
Teacher [43]	5.2	333.9	60.13
Ours: Student (1/2)	1.5	130.1	59.04
Ours: Student (1/4)	0.6	45.1	53.54

From Table 8, we observed quantitative comparisons of the computational complexity between the student network and the teacher network. Specifically, the student network, which reduced the channel by 1/4, decreased about 8.6 times compared to the parameters of the teacher, and FLOPS was reduced by 7.4 times. Above all, we should not overlook the fact that the performance of the student improved from 50.72% to 53.54% mAPH/L2 by our KD method. Furthermore, we apply our method to the student whose channel was reduced by half. The student's performance increases to 59.04%, and the parameters and FLOPS compared to the teacher are reduced by 3.5 times and 2.6 times, respectively. Compared to lightweight network-based methods [13] [35] [24] [46], our student networks are able to derive stable performance with fewer parameters and FLOPS in 3D object detection.

5. Conclusion

In this paper, we propose a novel KD method that transfers knowledge to produce a lightweight point cloud detector. Our main method involves interchange transfer, which learns coarse knowledge by increasing the similarity of the compressed feature and fine knowledge by decompressing the map-view feature of the other side using the channel-wise autoencoder. Moreover, we introduce a method to guide multiple detection head using head relation-aware self-attention, which refines knowledge by considering the relation of instances and properties. Ablation studies demonstrate the effectiveness of our proposed algorithm, and extensive experiments on the two large-scale open datasets verify that our proposed method achieves competitive performance against state-of-the-art methods.

Acknowledgement. This work was partly supported by NRF-2022R1A2C1091402, BK21 FOUR program of the NRF of Korea funded by the Ministry of Education (NRF5199991014091), and IITP grant funded by the Korea government(MSIT) (No.2021-0-00951, Development of Cloud based Autonomous Driving AI learning Software; No. 2021-0-02068, Artificial Intelligence Innovation Hub). W. Hwang is the corresponding author.

References

[1] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6

[2] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017. 2, 3, 5, 6, 7

[3] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9775–9784, 2019. 2

[4] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2021. 2, 8

[5] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017. 2

[6] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*, 2020. 2, 8

[7] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3153–3163, 2021. 3

[8] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 2

[9] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1, 3, 6, 8

[10] Zhen Huang, Xu Shen, Jun Xing, Tongliang Liu, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-Sheng Hua. Revisiting knowledge distillation: An inheritance and exploration framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3579–3588, 2021. 1, 3

[11] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018. 3

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 2, 3, 8

[14] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017. 2

[15] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 6356–6364, 2017. 1, 3

[16] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020. 3

[17] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, et al. Starnet: Targeted computation for object detection in point clouds. *arXiv preprint arXiv:1908.11069*, 2019. 2

[18] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[20] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 3, 6, 7

[21] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2397–2406, 2022. 2, 8

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 2

[23] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1, 2

[24] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020. 8

[25] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020. 2

[26] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 3

[27] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. 6

[28] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9395–9404, 2021. 1, 3

[29] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016. 2

[30] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 6

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[32] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019. 2, 3

[33] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020. 3

[34] Yue Wang and Justin M Solomon. Object dgcnn: 3d object detection using dynamic graphs. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 5, 6, 7

[35] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 8

[36] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 1, 2, 3

[37] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020. 3

[38] Jihan Yang, Shaoshuai Shi, Runyu Ding, Zhe Wang, and Xiaojuan Qi. Towards efficient 3d object detection with knowledge distillation. *arXiv preprint arXiv:2205.15156*, 2022. 3, 4, 6, 7, 8

[39] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 2

[40] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1951–1960, 2019. 2

[41] Maosheng Ye, Shuangjie Xu, and Tongyi Cao. Hvnet: Hybrid voxel network for lidar based 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1631–1640, 2020. 2

[42] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[43] 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. 1, 2, 3, 4, 6, 7, 8

[44] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *5th international conference on Learning Representations*, Apr. 2017. 1, 8

[45] Linfeng Zhang, Yukang Shi, Zuoqiang Shi, Kaisheng Ma, and Chenglong Bao. Task-oriented feature distillation. *Advances in Neural Information Processing Systems*, 33:14759–14771, 2020. 6, 7

[46] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. 8

[47] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021. 3, 6, 7

[48] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020. 2

[49] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 1, 2, 3