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

A Visualization of Important Points in PointRCNN Detectors

In order to have an intuitive understanding of feature values corresponding to object localization in the scene, we have also conducted the feature distribution visualization with PointRCNN on the KITTI dataset. As shown in Figure 1, the foreground object (*e.g.*, *cars* and *pedestrians*) features are more salient while the background points are suppressed, indicating that the our method can successfully localize the important points for not only voxel-based but also raw point-based detectors.

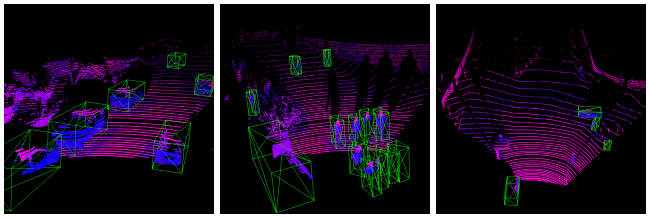

Figure 1: Visualization of feature distribution with PointRCNN on the KITTI dataset.

B Implementation Details

B.1 Implementation details on KITTI experiments

On PointPillars, the teacher network has 64 channels in the voxel encoder, 256 channels in the backbone network, and 384 channels in the head. The students with around 4×/16 × compression have 32/16 channels in their voxel encoders, 64/128 channels in their backbone networks, and 192/96 channels in their heads. On SECOND, the teacher network has 128 channels in the voxel encoder, 256 channels in the backbone network, and 512 channels in the head. The students with around 4×/16 × compression have 64/32 channels in their voxel encoders, 128/64 channels in their backbone networks, and 256/128 channels in their heads. On PointRCNN, the teacher network has 16-32-64-128-256-512 channels. The 4×/8× compression students have 8-16-32-64-128-256/5-11-22-45-91-182 channels, respectively. All the students are trained by 80 epochs with the AdamW optimizer. The learning rate is initialized as 1e-3 and decayed to 1e-7 gradually during training. L2-norm regularization and gradient clipping are also utilized to avoid overfitting and stabilize model training. Besides, random flipping, rotation, and point shuffle are utilized as data augmentations.

B.2 Implementation details of nuScenes experiments

On PointPillars, the teacher network has 64 channels in the voxel encoder, 256 channels in the backbone network, and 384 channels in the head. The students with around 2×/4 × compression have 46/32 channels in their voxel encoders, 182/128 channels in their backbone networks, and

Submitted to IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023). Do not distribute.

272/192 channels in their heads. On CenterPoint, the teacher network has 128 channels in the voxel encoder, 256 channels in the backbone network, and 256 channels in the head. The students with around 2×/4 × compression have 91/64 channels in their voxel encoders, 182/128 channels in their backbone networks, and 182/128 channels in their heads. Students are trained by 24 epochs with the AdamW optimizer. The learning rate is initialized as 1e-3, firstly linearly warmed up and then decayed by 10 at 20_{th} and 23_{th} epochs during training. L2-norm regularization and gradient clipping are also utilized to avoid overfitting and stabilize model training. Besides, random flipping, rotation, and point shuffle are utilized as data augmentations.

C More Experiments on Waymo Open Dataset (WOD) with CenterPoint

More experimental results on WOD are shown in Table 1. Experimental results on Waymo with CenterPoint-Pillar are shown in the following table. Note that we follow the default training setting in mmdetection3d by using 20% training data (Waymo-D5). We use the original CenterPoint model which is also trained with 20% Waymo data as the teacher and the CenterPoint models with fewer channels as the student. We follow the same training strategy by using the Adam optimizer and One-Cycle learning rate scheduler and train models for 36 epochs. It is observed that our method leads to **2.62** LEVEL_2 mAPH improvements, which outperforms the second-best method by **1.18** LEVEL_2 mAPH.

Table 1: Experimental results on Waymo Open Dataset with CenterPoint. A higher mAP and NDS indicate better performance.

Model	FLOPs (G)	#Params (M)	KD Method	LEVEL_2 mAPH(↑)
CenterPoint	224.6	5.2	Teacher w/o KD	59.10
	120.4	2.6	Student w/o KD	55.03
			+ Heo *et al.* [3]	56.47
			+ Tian *et al.* [5]	56.29
			+ Wang *et al.* [6]	56.15
			+ Ours	**57.65**

D Deployment and Training Efficiency

D.1 Latency Results

The latency of the student models in our method are shown in Table 2. These results are measured on single 2080Ti GPU with TensorRT. It is observed that: (i) On PointPillars, our student leads to 0.9 and 1.8 mAP improvements on BEV and 3D detection and 2.2× acceleration on latency. (ii) On SECOND, our student leads to 0.9 and 0.1 mAP improvements on BEV and 3D detection and 2.5× acceleration on latency. (iii) On PointRCNN, our student achieves 3.4 × latency acceleration with only 0.2 and 1.3 mAP drop on BEV and 3D detection, respectively. These observations indicates that the effectiveness of our method is still significant in terms of latency reduction.

Table 2: Comparison on the latency between students and teachers on KITTI.

Model	FLOPs (G)	#Params (M)	Latency (ms)	BEV mAP	3D mAP
PointPillars Teacher	34.3	4.8	20.3	68.3	60.3
PointPillars Student	9.0	1.3	9.2	69.2	62.1
SECOND Teacher	69.8	5.3	35.1	72.3	66.1
SECOND Student	17.8	1.4	14.3	73.2	66.2
PointRCNN Teacher	104.9	4.1	178.6	75.3	70.9
PointRCNN Student	13.7	0.5	52.2	75.1	69.6

D.2 Training Costs

Our method does not introduce significant additional training costs. Taking PointPillars on KITTI as an example, the student trained with naïve feature-based knowledge distillation and the student trained with our method requires 10.01 and 10.04 GPU hours, indicating our method does not leads to much more training costs than traditional knowledge distillation methods. Note that since the KNN operation and dynamic graph convolution in our method is applied to features of the voxels or downsampled points instead of the raw inputs, local distillation in our method has good training efficiency. KNN in our method is scalable to large-scale datasets and it does not lead to significant computation overhead. In voxels-based detectors, KNN in our method is applied to the features of each voxel instead of each point. Hence, the increment in the number of points directly increases the computation complexity of KNN. In raw points-based detectors, KNN in our method is applied to the feature of downsampled points instead of the inputted points. Hence its computation is ignorable compared with the computation in the backbone network. Measured with 2080Ti GPU on Waymo with CenterPoint, our experimental results show that KNN in our method account for less than 1% training overhead. Besides, KNN has been well supported by modern hardware acceleration toolkits such as CUDA. Moreover, since our method is only utilized during the training period, it does not introduce additional computation during inference.

E Experiments on Model Compression with Layer Reduction Strategy

In order to achieve high-efficiency deployment of the 3D detectors, there are generally three kinds of model compression strategies. The commonly used strategies include reducing the number of model channels or depths (*i.e.*, layers), and quantizing the weights and activations to shortened bit-width. In this paper, we mainly use the channel reduction strategy. As shown in Table 2, this compression strategy is simple and effective both to compress the model and speed up the model inference. We have also conducted comprehensive experiments on depth reduction by using different layer configurations of the SECOND backbone model in PointPillars on the KITTI dataset. The results are shown in Table 3, in which the *Number of Layers* indicates the number of layers in each stage. For example, "3+5+5=15" denotes a total number of fifteen layers with three, five, and five layers for the first, second, and third stages, respectively. The results demonstrate that our method significantly improves the models with reduced depth, consistently surpassing the teacher model in both 3D and BEV mAP by a large margin. This evaluation strongly supports the potential of our PointDistiller in model compression with other strategies, and we believe it is promising to extend our method to strategies like quantization other than channel/depth reduction.

Table 3: Experimental results on students which have the same channels but fewer layers than their teachers. The model in the first row indicates the teacher model.

Number of Layers	FLOPs (G)	#Params (M)	KD	BEV mAP(↑)	3D mAP(↑)
3+5+5=13 (Teacher)	34.3	4.8	×	68.3	60.3
2+3+3=8	24.4	3.3	×	67.8	60.1
2+3+3=8	24.4	3.3	✓	69.5	62.6
1+2+2=5	20.5	2.6	×	67.3	59.8
1+2+2=5	20.5	2.6	✓	69.3	62.3
1+1+1=3	14.5	1.8	×	67.1	59.2
1+1+1=3	14.5	1.8	✓	69.1	62.0

Broader Impact

This paper proposes a novel knowledge distillation method named PointDistiller to compress the point clouds-based 3D detectors. It can accelerate and reduce the energy cost for object detection in multiple applications and has no potential social impact.

Limitations and Future Works

In this paper, we mainly evaluate our method on point cloud 3D detection on KITTI [2], nuScene [1] and Waymo [4]. We will conduct more experiments for images and point clouds based multi-modal detection on more datasets in the future.

F Comparison with More Methods

BEV Detection				
KD Method	Car	Ped.	Cyc.	mAP
Student w/o KD	88.1	51.8	65.0	68.3
Yang *et al.*, NeurIPS'22	88.3	52.2	65.1	68.5
Hou *et al.*, CVPR'22	88.3	52.1	65.5	68.6
Ju *et al.*, ACM MM'22	88.2	51.9	65.2	68.4
Ours	**89.0**	**52.8**	**65.8**	**69.2**

3D Detection				
KD Method	Car	Ped.	Cyc.	mAP
Student w/o KD	75.9	43.0	57.2	58.7
Yang *et al.*, NeurIPS'22	76.3	45.8	58.0	60.0
Hou *et al.*, CVPR'22	76.2	45.1	58.3	59.9
Ju *et al.*, ACM MM'22	75.4	46.2	59.1	60.2
Ours	**76.9**	**47.5**	**62.0**	**62.1**

Table 4: Comparison experiments with PointPillars on KITTI.

We have added comparison with Yang *et al.* [**?**], Hou *et al.* [**?**] and Ju *et al.* [**?**] . As shown in 4, our method outperforms the second-best method by **0.6** mAP and **1.9** mAP on BEV and 3D detection.

References

[1] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 11618–11628. Computer Vision Foundation / IEEE, 2020. 4

[2] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4

[3] Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., and Choi, J. Y. A comprehensive overhaul of feature distillation. In *Int. Conf. Comput. Vis. (ICCV)*, pp. 1921–1930, 2019. 2

[4] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 2443–2451. Computer Vision Foundation / IEEE, 2020. 4

[5] Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2020. 2

[6] Wang, Y., Fathi, A., Wu, J., Funkhouser, T. A., and Solomon, J. Multi-frame to single-frame: Knowledge distillation for 3d object detection. *CoRR*, abs/2009.11859, 2020. URL `https://arxiv.org/abs/2009.11859`. 2