Towards Efficient 3D Object Detection with Knowledge Distillation

Jihan Yang, Shaoshuai Shi, Runyu Ding, Zhe Wang, and XIAOJUAN QI

NeurIPS 2022

Efficient Deep Learning Reading Group

Presenter: Pengcheng Wang

Content

Background

Background: 3D Object Detection

Sensors:

Camera: RGB-Depth, **LiDAR: Light Detection And Ranging**, Radar: Radio Detection and Ranging

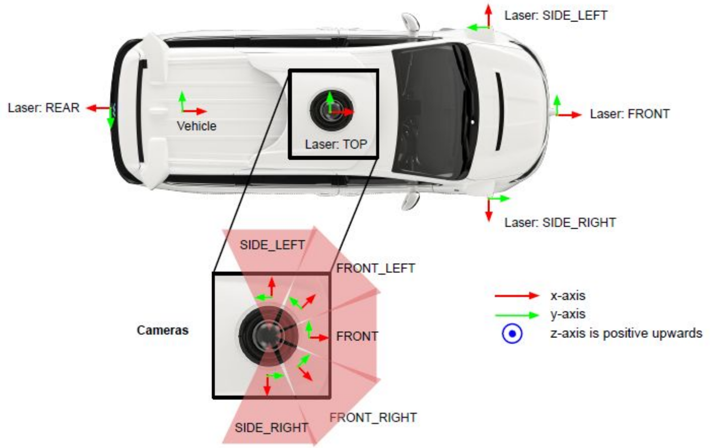
2D vs 3D:

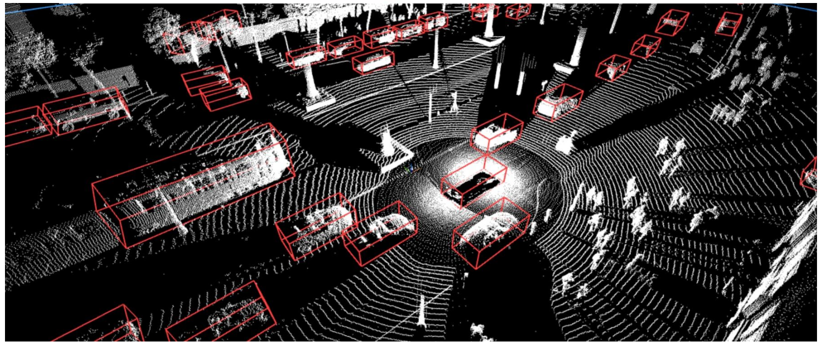
3D: 7 Degrees of freedom (DoF) ~ 9 DoF

Center [x, y, z] + Object Size [length, width, height] + Rotation: [yaw, roll, pitch]

2D: 4 DoF

Center [x, y] + Object Size [width, height]





Background: 3D Object Detection Methods

Point-based method:

Directly processing the point clouds using deep learning networks to classify and regress objects. This method has gained popularity due to its ability to handle the irregularity and sparsity of point cloud data. PointNet, PointNet++, and Frustum PointNets are some of the popular point-based methods used for 3D object detection.

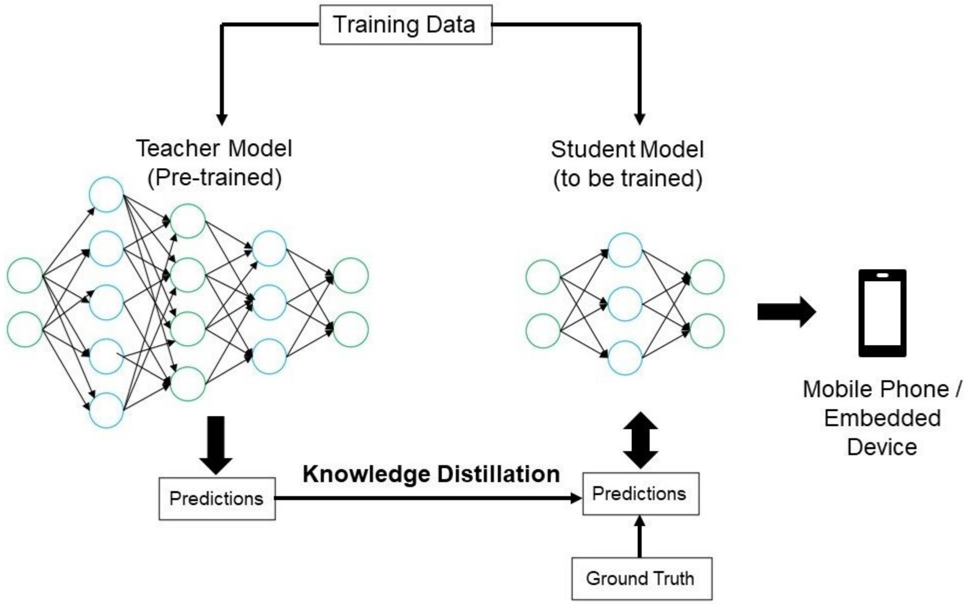
Grid-based method:

The grid-based method **converts the point cloud data into a 3D grid structure**, where each cell in the grid represents a voxel. The presence of an object within a voxel is determined by the number of points falling inside the voxel. The grid-based method allows for the use of 2D and 3D convolutional neural networks (CNNs) for object detection, making it computationally efficient. Some popular grid-based methods include VoxelNet and SECOND.

Graph-based method:

The graph-based method converts the point cloud data into a graph structure, where each point is represented as a node in the graph, and the edges between the nodes represent their spatial relationships. The graph-based method uses graph neural networks (GNNs) to process the graph structure for object detection. The graph-based method is effective in detecting objects with complex and irregular shapes, making it suitable for various applications.

Background: Knowledge Distillation

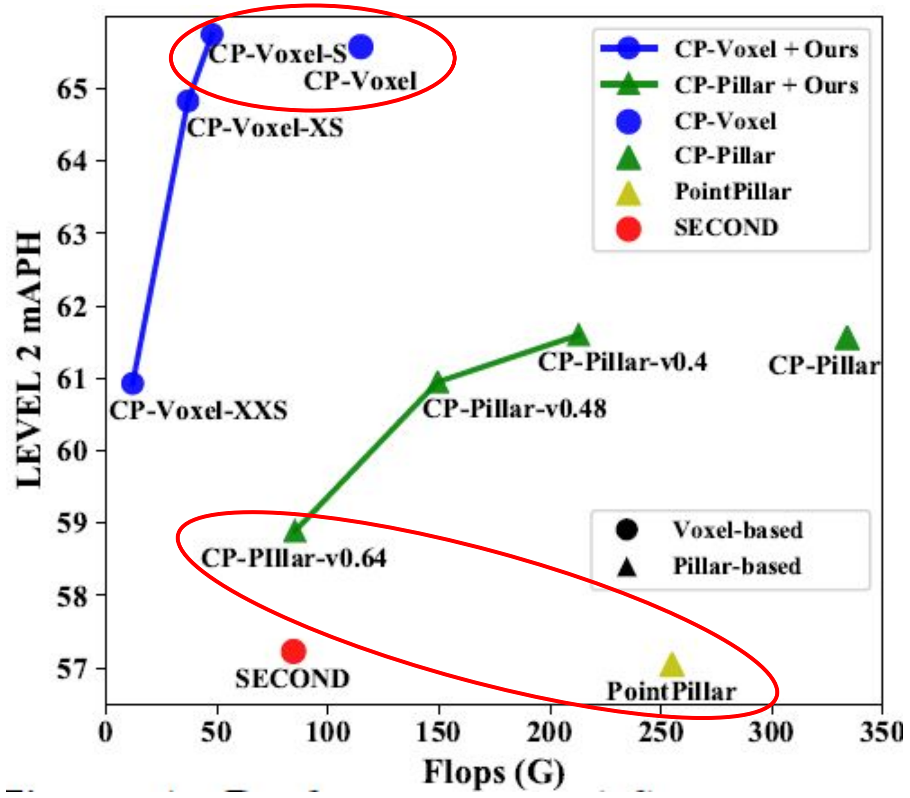

Training Data
Teacher Model (Pre-trained)
Student Model (to be trained)
Mobile Phone / Embedded Device
Predictions
Knowledge Distillation
Predictions
Ground Truth

Knowledge Distillation is a technique in deep learning where a smaller neural network, called the student, is trained to mimic the behavior of a larger, more complex neural network, called the teacher. The goal is to transfer the knowledge and generalization power of the teacher network to the student network, making it more compact and computationally efficient.

Contributions:

1. First, the authors study how to obtain lightweight student detectors with satisfactory efficiency and accuracy trade offs given a pre-trained teacher 3D object detector.
2. Second, the authors empirically investigate the effectiveness of existing knowledge distillation methods on this new setting upon accurate teacher models and efficient student models.
3. Third, the authors propose simple, general, and effective strategies to improve knowledge distillation on 3D object detection upon the strong KD baseline derived as above.
4. Finally, the authors’ empirical studies on efficient model design and knowledge distillation methods yield superior performance in delivering efficient and effective pillar- and voxel-based 3D detectors.

Teaser Results



Teacher Model: CP-Voxel, CP-Pillar

Best performing student model: CP-Voxel-S

Most efficient pillar-based model: CP-Pillar-v0.64

Baselines: SECOND, PointPillar

Designing Efficient Student Networks

Basic Setups and Evaluation Metrics

Models: CenterPoint-Pillar (CP-Pillar) and CenterPoint-Voxel (CP-Voxel) *[CVPR 2021]*

Dataset: on the largest annotated 3D LiDAR perception dataset Waymo Open Dataset (WOD)

Metrics:

- number of parameters, flops, activations, latency (i.e. test time) and peak GPU training memory (batch size 1) as quantitative indicators to evaluate model efficiency from parameter, computation and memory throughout aspects
- LEVEL 2 mAPH as the performance evaluation metric following WOD

Quantitative indicator: Cost Performance Ratio (CPR)

$$\mathrm{CPR} = 0.5 \times (1 - \frac{\mathrm{acts}_s}{\mathrm{acts}_t}) + 0.5 \times (\frac{\mathrm{mAPH}_s}{\mathrm{mAPH}_t})^3, \tag{1}$$

Acceleration Strategy

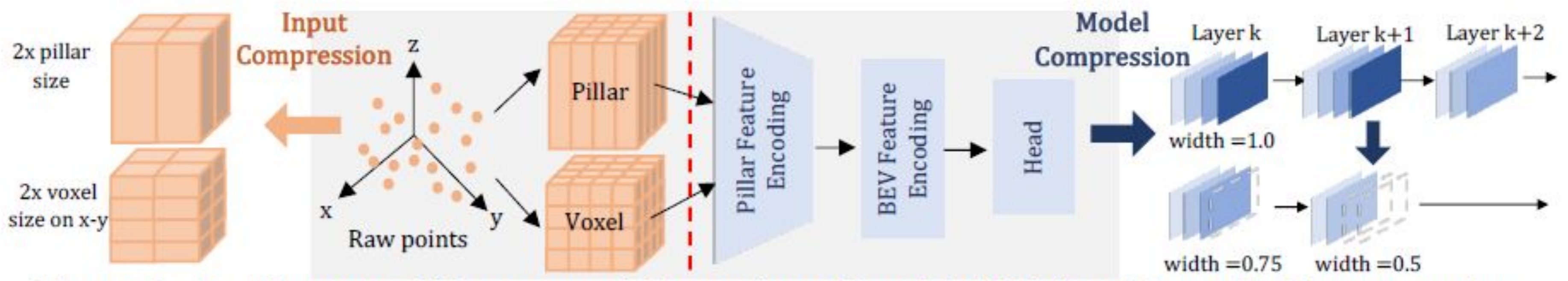


Figure 2: Architecture of detector and illustration of model (right) and input (left) compression.

- Model compression:
  - Trimming along depth (i.e.number of layers) or width (number of channels)
  - 3 major modules: Pillar Feature Encoding (PFE) module, Bird eye's view Feature Encoding (BFE) module and detection head
- Input resolution compression:
  - Increase the voxel/pillar size on the x-y plane when constructing voxel/pillar

Model compression results

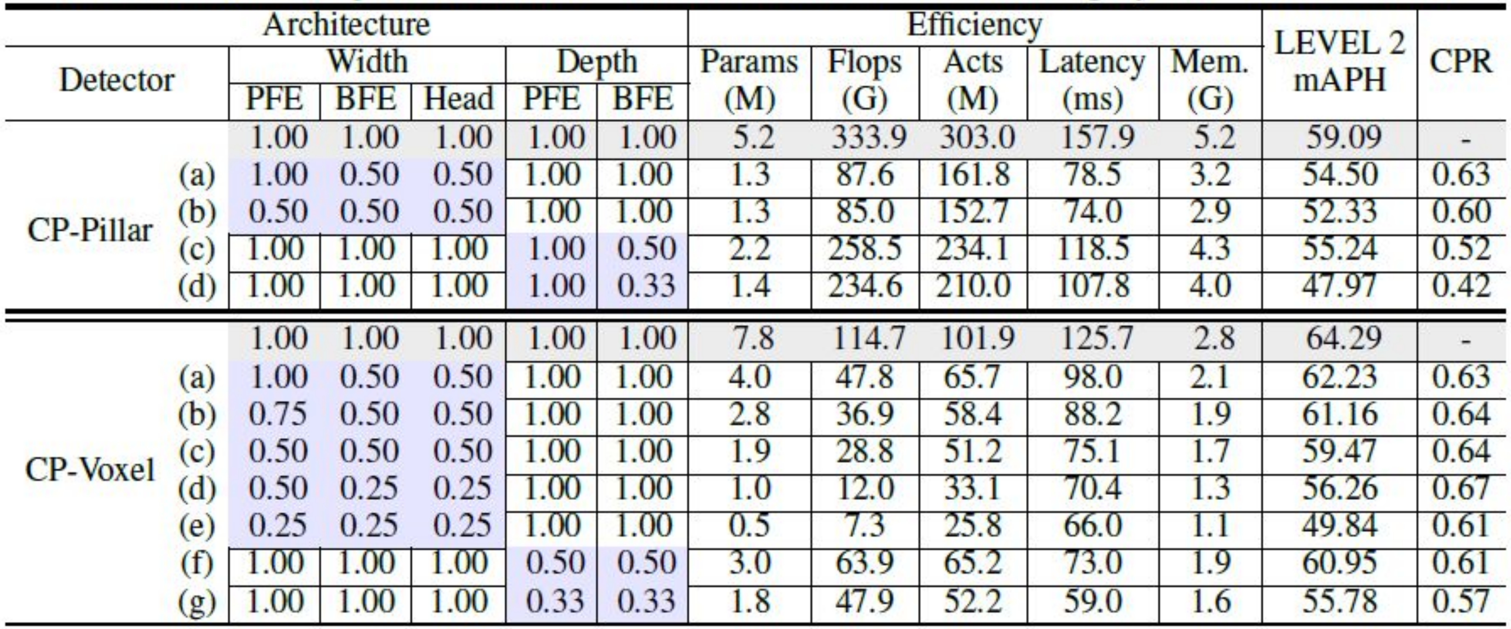


Table 1: Model compression results. Teacher models are marked in gray. See text for details.

Detector		Architecture					Efficiency					LEVEL 2 mAPH	CPR
		Width			Depth		Params (M)	Flops (G)	Acts (M)	Latency (ms)	Mem. (G)		
		PFE	BFE	Head	PFE	BFE							
CP-Pillar		1.00	1.00	1.00	1.00	1.00	5.2	333.9	303.0	157.9	5.2	59.09	-
	(a)	1.00	0.50	0.50	1.00	1.00	1.3	87.6	161.8	78.5	3.2	54.50	0.63
	(b)	0.50	0.50	0.50	1.00	1.00	1.3	85.0	152.7	74.0	2.9	52.33	0.60
	(c)	1.00	1.00	1.00	1.00	0.50	2.2	258.5	234.1	118.5	4.3	55.24	0.52
	(d)	1.00	1.00	1.00	1.00	0.33	1.4	234.6	210.0	107.8	4.0	47.97	0.42
CP-Voxel		1.00	1.00	1.00	1.00	1.00	7.8	114.7	101.9	125.7	2.8	64.29	-
	(a)	1.00	0.50	0.50	1.00	1.00	4.0	47.8	65.7	98.0	2.1	62.23	0.63
	(b)	0.75	0.50	0.50	1.00	1.00	2.8	36.9	58.4	88.2	1.9	61.16	0.64
	(c)	0.50	0.50	0.50	1.00	1.00	1.9	28.8	51.2	75.1	1.7	59.47	0.64
	(d)	0.50	0.25	0.25	1.00	1.00	1.0	12.0	33.1	70.4	1.3	56.26	0.67
	(e)	0.25	0.25	0.25	1.00	1.00	0.5	7.3	25.8	66.0	1.1	49.84	0.61
	(f)	1.00	1.00	1.00	0.50	0.50	3.0	63.9	65.2	73.0	1.9	60.95	0.61
	(g)	1.00	1.00	1.00	0.33	0.33	1.8	47.9	52.2	59.0	1.6	55.78	0.57

- **Width vs. depth compression**: width-level pruning is preferred
- **Module-wise pruning selection**: PFE module has the least redundancy to be reduced
- **Favorable compression strategies for different detection architectures**: pillar-based architecture (i.e. CP-Pillar) is more suitable for input compression while voxel-based architecture (i.e. CP-Voxel) prefers width level compression.

Input resolution compression results

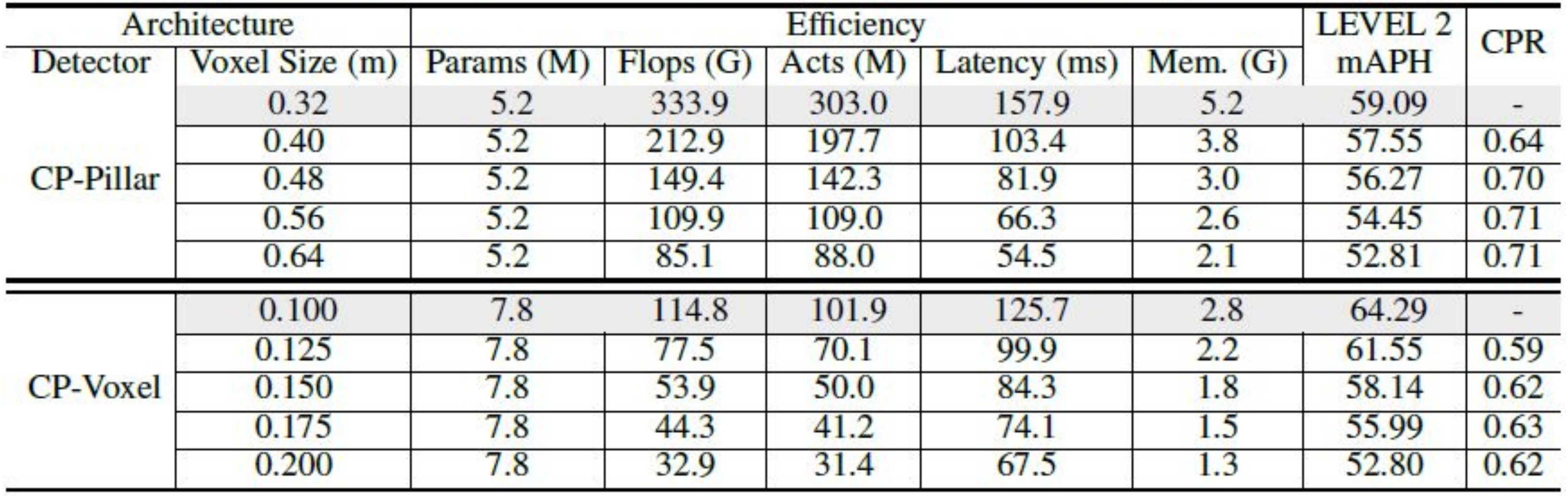


Table 2: Input compression results. Teacher models are marked in gray. See text for details.

Architecture		Efficiency					LEVEL 2	CPR
Detector	Voxel Size (m)	Params (M)	Flops (G)	Acts (M)	Latency (ms)	Mem. (G)	mAPH	
CP-Pillar	0.32	5.2	333.9	303.0	157.9	5.2	59.09	-
	0.40	5.2	212.9	197.7	103.4	3.8	57.55	0.64
	0.48	5.2	149.4	142.3	81.9	3.0	56.27	0.70
	0.56	5.2	109.9	109.0	66.3	2.6	54.45	0.71
	0.64	5.2	85.1	88.0	54.5	2.1	52.81	0.71
CP-Voxel	0.100	7.8	114.8	101.9	125.7	2.8	64.29	-
	0.125	7.8	77.5	70.1	99.9	2.2	61.55	0.59
	0.150	7.8	53.9	50.0	84.3	1.8	58.14	0.62
	0.175	7.8	44.3	41.2	74.1	1.5	55.99	0.63
	0.200	7.8	32.9	31.4	67.5	1.3	52.80	0.62

By halving the input resolution, the computation overhead for 2D convolution layers in the BFE module and detection head will be reduced to ¼

Favorable compression strategies for different detection architectures: pillar-based architecture (i.e. CP-Pillar) is more suitable for input compression while voxel-based architecture (i.e. CP-Voxel) prefers width level compression

Benchmark Knowledge Distillation for 3D Object Detection

Benchmark Knowledge Distillation for 3D Object Detection

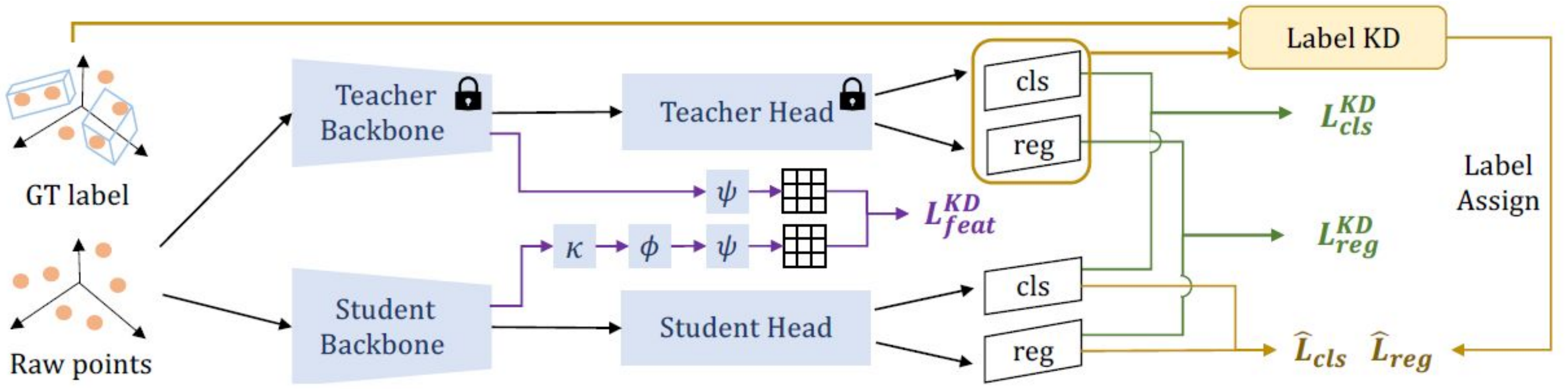


Figure 3: Overall KD paradigm for 3D detection. Teacher weights are frozen during the whole distillation procedure. Logit, feature and label KD are colored by green, purple and yellow, respectively.

Benchmark seven popular 2D KD methods including **logit KD** (i.e. KD and GID-L), **feature KD** (i.e. FitNet, Mimic, FG and GID-F) and **label KD** on six teacher-student pairs with comprehensive analysis.

Logit KD

Takes teacher model’s final response as guidance for training a student network and is closely related to the specific task. In 3D object detection, we calculate the logit KD loss between teacher and student outputs as follows:

$$\mathcal{L}_{\mathrm{cls}}^{\mathrm{KD}} = \mathbb{E}[m_{\mathrm{cls}} \| \kappa(p_{\mathrm{cls}}^{s}) - p_{\mathrm{cls}}^{t} \|_2], \qquad \mathcal{L}_{\mathrm{reg}}^{\mathrm{KD}} = \mathcal{L}_{\mathrm{reg}}(p_{\mathrm{reg}}^{s},\ p_{\mathrm{reg}}^{t}), \qquad (2)$$

where superscripts **s** and **t** indicate student and teacher, **pcls** and **preg** represent the classification response after the sigmoid and bounding box regression prediction of detector separately, **κ** is the bilinear interpolation to match student output resolutions towards teacher, **Lreg** is the regression loss function of 3D detector and **mcls** is a mask ranged in [0, 1] to indicate important regions in pcls.

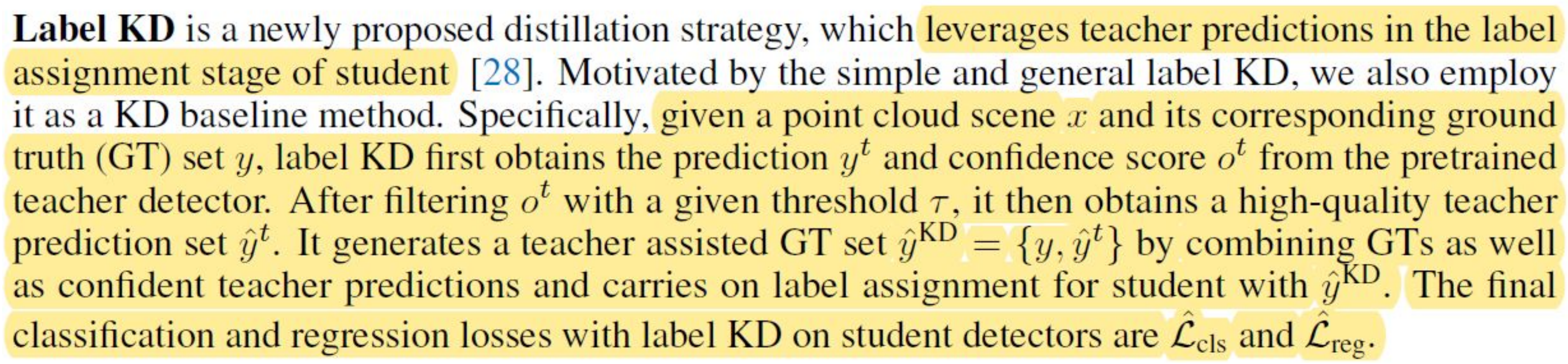
Feature KD

Enforces student models to mimic teacher models' intermediate feature maps. Specifically, we construct feature mimicking on the last layer of BFE between student and teacher network as follows:

$$\mathcal{L}_{\text{feat}}^{\text{KD}} = \mathbb{E}[m_{\text{feat}} \| \psi(\phi(\kappa(f^s)), y) - \psi(f^t, y) \|_2], \tag{3}$$

where **y** is the ground truth, **ψ** indicates the RoI Align, **ϕ** is a 1 × 1 convolution with batch normalization and ReLU block to align channel-wise discrepancy between teacher feature **ft** and student feature **fs**, **mfeat** is the mask to indicate critical regions ranged in [0, 1].

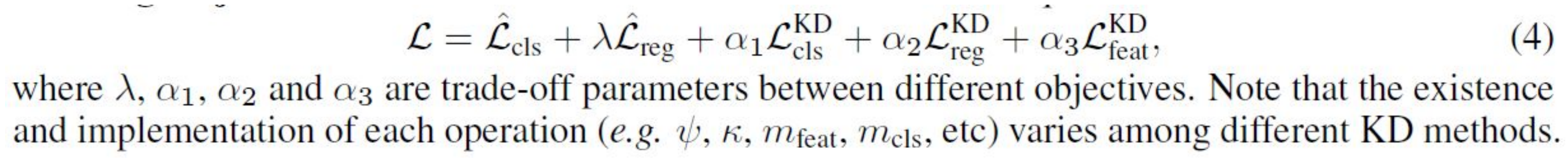
Label KD



Label KD is a newly proposed distillation strategy, which leverages teacher predictions in the label assignment stage of student [28]. Motivated by the simple and general label KD, we also employ it as a KD baseline method. Specifically, given a point cloud scene x and its corresponding ground truth (GT) set y, label KD first obtains the prediction y^t and confidence score o^t from the pretrained teacher detector. After filtering o^t with a given threshold τ, it then obtains a high-quality teacher prediction set $\hat{y}^t$. It generates a teacher assisted GT set $\hat{y}^{\mathrm{KD}} = \{y, \hat{y}^t\}$ by combining GTs as well as confident teacher predictions and carries on label assignment for student with $\hat{y}^{\mathrm{KD}}$. The final classification and regression losses with label KD on student detectors are $\hat{\mathcal{L}}_{\mathrm{cls}}$ and $\hat{\mathcal{L}}_{\mathrm{reg}}$.

Training Objective

Combination of three stream KD techniques as follows:



$$\mathcal{L} = \hat{\mathcal{L}}_{\text{cls}} + \lambda\hat{\mathcal{L}}_{\text{reg}} + \alpha_1\mathcal{L}^{\text{KD}}_{\text{cls}} + \alpha_2\mathcal{L}^{\text{KD}}_{\text{reg}} + \alpha_3\mathcal{L}^{\text{KD}}_{\text{feat}}, \tag{4}$$

where λ, α_1, α_2 and α_3 are trade-off parameters between different objectives. Note that the existence and implementation of each operation (*e.g.* ψ, κ, m_{feat}, m_{cls}, etc) varies among different KD methods.

Results and Analysis

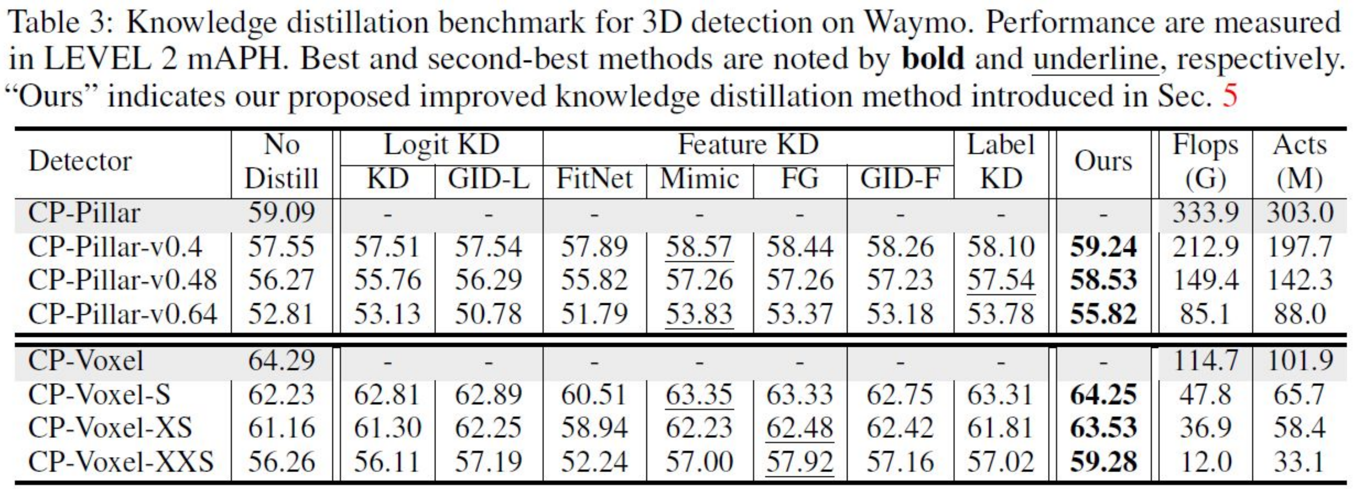


Table 3: Knowledge distillation benchmark for 3D detection on Waymo. Performance are measured in LEVEL 2 mAPH. Best and second-best methods are noted by **bold** and underline, respectively. "Ours" indicates our proposed improved knowledge distillation method introduced in Sec. 5

Detector	No Distill	Logit KD		Feature KD				Label KD	Ours	Flops (G)	Acts (M)
		KD	GID-L	FitNet	Mimic	FG	GID-F				
CP-Pillar	59.09	-	-	-	-	-	-	-	-	333.9	303.0
CP-Pillar-v0.4	57.55	57.51	57.54	57.89	58.57	58.44	58.26	58.10	**59.24**	212.9	197.7
CP-Pillar-v0.48	56.27	55.76	56.29	55.82	57.26	57.26	57.23	57.54	**58.53**	149.4	142.3
CP-Pillar-v0.64	52.81	53.13	50.78	51.79	53.83	53.37	53.18	53.78	**55.82**	85.1	88.0
CP-Voxel	64.29	-	-	-	-	-	-	-	-	114.7	101.9
CP-Voxel-S	62.23	62.81	62.89	60.51	63.35	63.33	62.75	63.31	**64.25**	47.8	65.7
CP-Voxel-XS	61.16	61.30	62.25	58.94	62.23	62.48	62.42	61.81	**63.53**	36.9	58.4
CP-Voxel-XXS	56.26	56.11	57.19	52.24	57.00	57.92	57.16	57.02	**59.28**	12.0	33.1

- Feature-based KD methods (i.e. Mimic and FG) achieve prominent performance, which demonstrates the strong potential of learning from teacher's hints on feature extraction.
- Furthermore, we find that instance-aware local region imitation is important in distillation for 3D detection, as enormous background regions overwhelm the supervision of sparse instances. For instance, with instance-aware imitation, Mimic, FG and GID-F consistently outperform FitNet which fully imitates all spatial positions of teacher feature maps. Similar conclusions can also be drawn in logit KD by comparing the results of instance-aware GID-L and vanilla KD.

Synergy Analysis

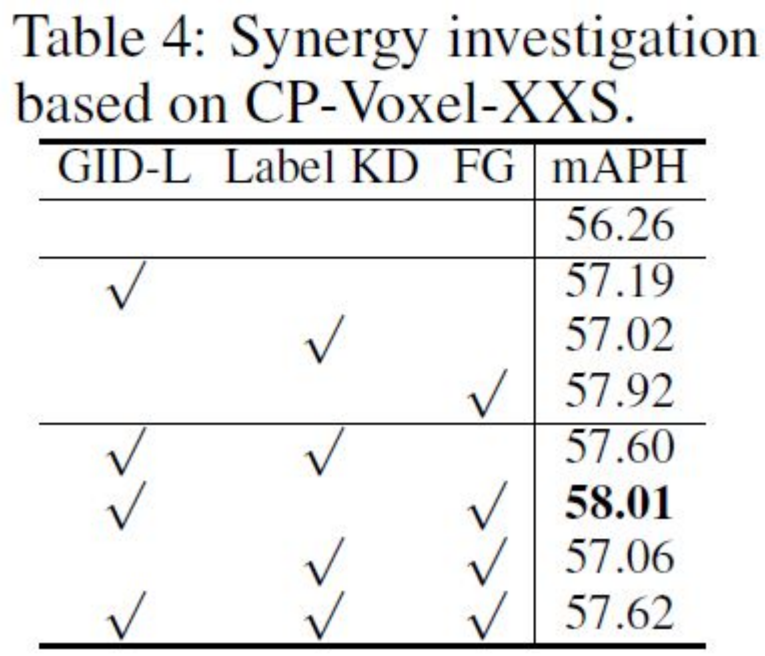


Table 4: Synergy investigation based on CP-Voxel-XXS.

GID-L	Label KD	FG	mAPH
			56.26
√			57.19
	√		57.02
		√	57.92
√	√		57.60
√		√	**58.01**
	√	√	57.06
√	√	√	57.62

Although feature KD itself achieves the highest performance on CP-Voxel-XXS compared to logit KD and label KD techniques, it can hardly achieve improvements or even suffers from performance degradation when combined with other KD methods. On the contrary, logit KD and label KD can cooperate well with each other to further improve student's capability.

This is potentially caused by logit KD and label KD implicitly enforcing regularization on the feature, which can be conflicted with the optimization direction of feature KD and results in a poor synergy effect.

Improved Knowledge Distillation for 3D Object Detection

Pivotal Position Logit KD

Motivated by the imbalance of foreground and background regions, previous 2D methods attempt to only enforce output-level imitation on pixels near or covering instances.

However, we find that it is sub-optimal in 3D scenarios given more extreme imbalance between small informative instances and large redundant background areas. For example, based on CP-Pillar, even a vehicle with 10m length and 4m width occupies only 32×13 pixels in the final 468×468 response map. **Such small instances and large perception ranges in 3D detection requires more sophisticated imitation region selection than previous coarse instance-wise masking manners in 2D detection.** Hence, we propose **Pivotal Position (PP) logit KD** which leverages cues in teacher classification response or label assignment to determine the important areas for distillation.

Teacher Guided Initialization

Directly use the trained weights of teacher to serve as the initialization of student network, named teacher guided initialization (TGI) to enhance student model's feature extraction abilities by inheriting it from a teacher model.

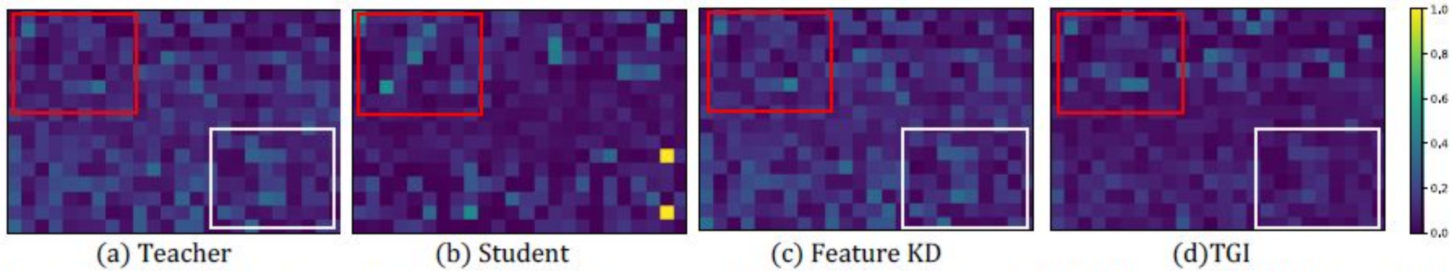


Figure 4: Visualization of channel-wise L_1 norm for distillation from CP-Pillar to CP-Pillar-v0.64.

Main Results and Comparison

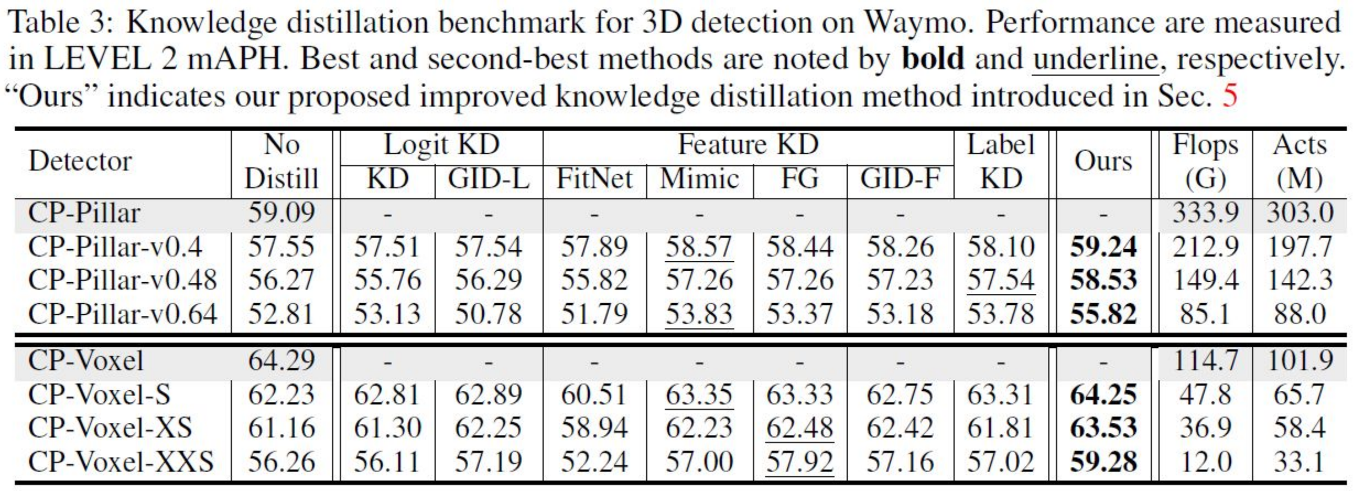


Table 3: Knowledge distillation benchmark for 3D detection on Waymo. Performance are measured in LEVEL 2 mAPH. Best and second-best methods are noted by **bold** and underline, respectively. "Ours" indicates our proposed improved knowledge distillation method introduced in Sec. 5

Detector	No Distill	Logit KD		Feature KD				Label KD	Ours	Flops (G)	Acts (M)
		KD	GID-L	FitNet	Mimic	FG	GID-F				
CP-Pillar	59.09	-	-	-	-	-	-	-	-	333.9	303.0
CP-Pillar-v0.4	57.55	57.51	57.54	57.89	58.57	58.44	58.26	58.10	**59.24**	212.9	197.7
CP-Pillar-v0.48	56.27	55.76	56.29	55.82	57.26	57.26	57.23	57.54	**58.53**	149.4	142.3
CP-Pillar-v0.64	52.81	53.13	50.78	51.79	53.83	53.37	53.18	53.78	**55.82**	85.1	88.0
CP-Voxel	64.29	-	-	-	-	-	-	-	-	114.7	101.9
CP-Voxel-S	62.23	62.81	62.89	60.51	63.35	63.33	62.75	63.31	**64.25**	47.8	65.7
CP-Voxel-XS	61.16	61.30	62.25	58.94	62.23	62.48	62.42	61.81	**63.53**	36.9	58.4
CP-Voxel-XXS	56.26	56.11	57.19	52.24	57.00	57.92	57.16	57.02	**59.28**	12.0	33.1

The improved KD pipeline consistently surpasses previous KD strategies on all settings with 0.7% ~ 1.9% improvement, thanks to our enhanced logit KD and more collaborative TGI.

Comparison with Other Detectors

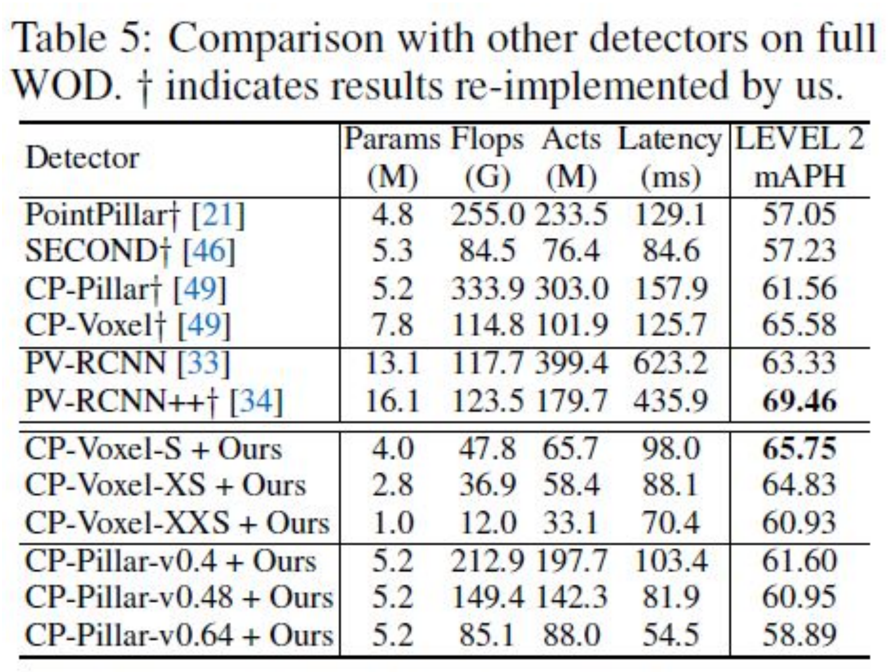


Table 5: Comparison with other detectors on full WOD. † indicates results re-implemented by us.

Detector	Params (M)	Flops (G)	Acts (M)	Latency (ms)	LEVEL 2 mAPH
PointPillar† [21]	4.8	255.0	233.5	129.1	57.05
SECOND† [46]	5.3	84.5	76.4	84.6	57.23
CP-Pillar† [49]	5.2	333.9	303.0	157.9	61.56
CP-Voxel† [49]	7.8	114.8	101.9	125.7	65.58
PV-RCNN [33]	13.1	117.7	399.4	623.2	63.33
PV-RCNN++† [34]	16.1	123.5	179.7	435.9	**69.46**
CP-Voxel-S + Ours	4.0	47.8	65.7	98.0	**65.75**
CP-Voxel-XS + Ours	2.8	36.9	58.4	88.1	64.83
CP-Voxel-XXS + Ours	1.0	12.0	33.1	70.4	60.93
CP-Pillar-v0.4 + Ours	5.2	212.9	197.7	103.4	61.60
CP-Pillar-v0.48 + Ours	5.2	149.4	142.3	81.9	60.95
CP-Pillar-v0.64 + Ours	5.2	85.1	88.0	54.5	58.89

With similar latency and much fewer parameters, flops as well as activations, CP-Voxel-XS outperforms SECOND by 7.6%. Our CP-Pillar-v0.64, is 2.4× faster than PointPillar on a GTX-1060 while achieves 1.8% higher performance.

Cross Stage Distillation

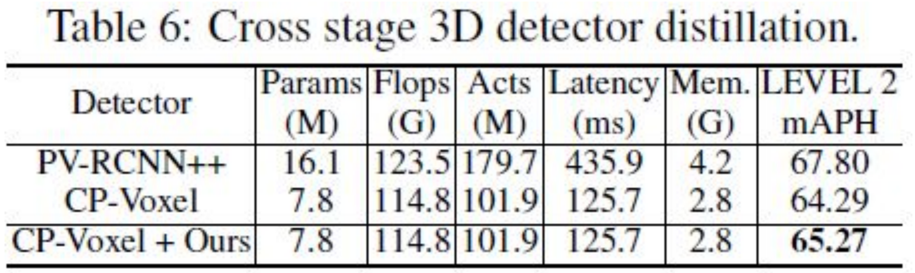


Table 6: Cross stage 3D detector distillation.

Detector	Params (M)	Flops (G)	Acts (M)	Latency (ms)	Mem. (G)	LEVEL 2 mAPH
PV-RCNN++	16.1	123.5	179.7	435.9	4.2	67.80
CP-Voxel	7.8	114.8	101.9	125.7	2.8	64.29
CP-Voxel + Ours	7.8	114.8	101.9	125.7	2.8	**65.27**

However, despite performance improvements by 3.5%, PVRCNN++ requires around 2.2× parameters, 1.8× activations and 3.5× latency compared to CP-Voxel.

Leveraging hints from pretrained PVRCNN++, our distilled CP-Voxel achieves around 1% performance gains without any extra computation and parameter overheads during inference.

Generalization to More Scenarios

Generality on Other Dataset and Detector

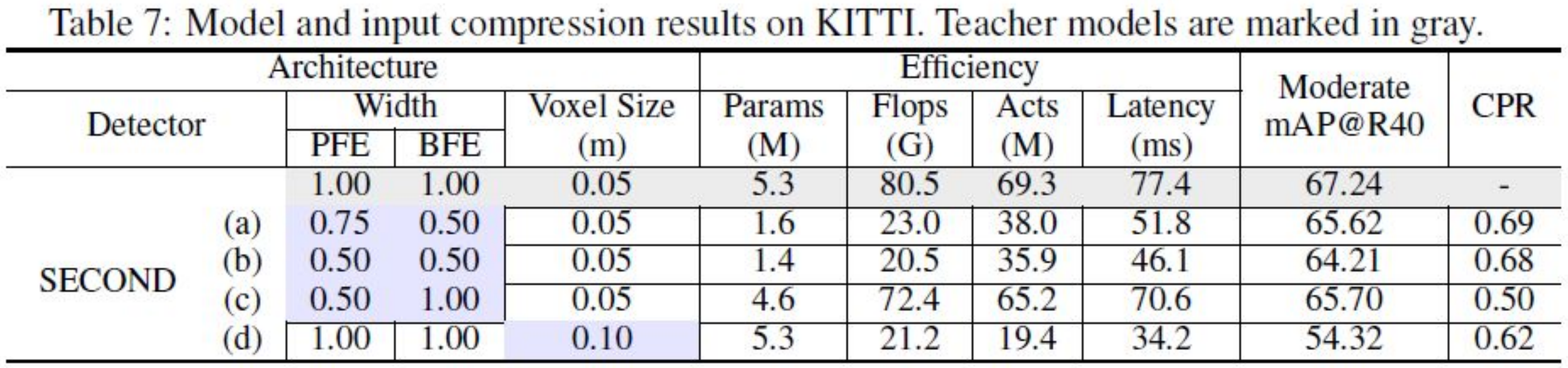


Table 7: Model and input compression results on KITTI. Teacher models are marked in gray.

Architecture					Efficiency				Moderate mAP@R40	CPR
Detector		Width		Voxel Size	Params	Flops	Acts	Latency		
		PFE	BFE	(m)	(M)	(G)	(M)	(ms)		
SECOND		1.00	1.00	0.05	5.3	80.5	69.3	77.4	67.24	-
	(a)	0.75	0.50	0.05	1.6	23.0	38.0	51.8	65.62	0.69
	(b)	0.50	0.50	0.05	1.4	20.5	35.9	46.1	64.21	0.68
	(c)	0.50	1.00	0.05	4.6	72.4	65.2	70.6	65.70	0.50
	(d)	1.00	1.00	0.10	5.3	21.2	19.4	34.2	54.32	0.62

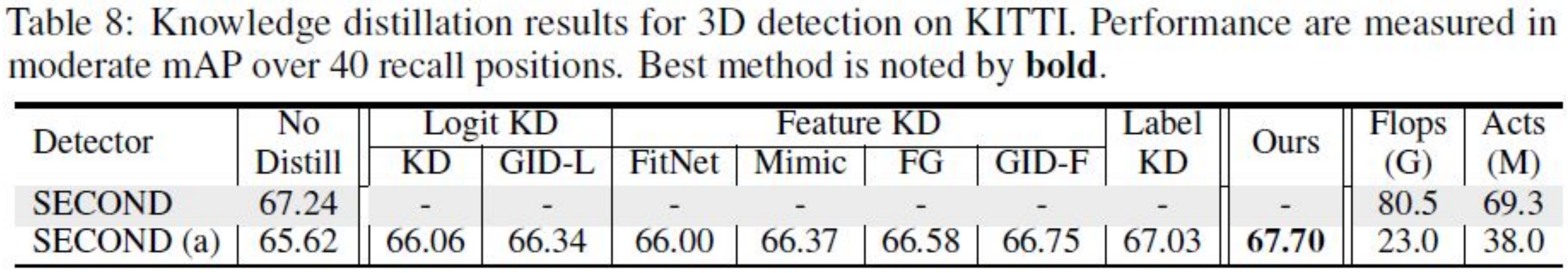


Table 8: Knowledge distillation results for 3D detection on KITTI. Performance are measured in moderate mAP over 40 recall positions. Best method is noted by **bold**.

Detector	No Distill	Logit KD		Feature KD				Label KD	Ours	Flops (G)	Acts (M)
		KD	GID-L	FitNet	Mimic	FG	GID-F				
SECOND	67.24	-	-	-	-	-	-	-	-	80.5	69.3
SECOND (a)	65.62	66.06	66.34	66.00	66.37	66.58	66.75	67.03	**67.70**	23.0	38.0

Generalization to More Scenarios

Generality on Advance Compression Manner

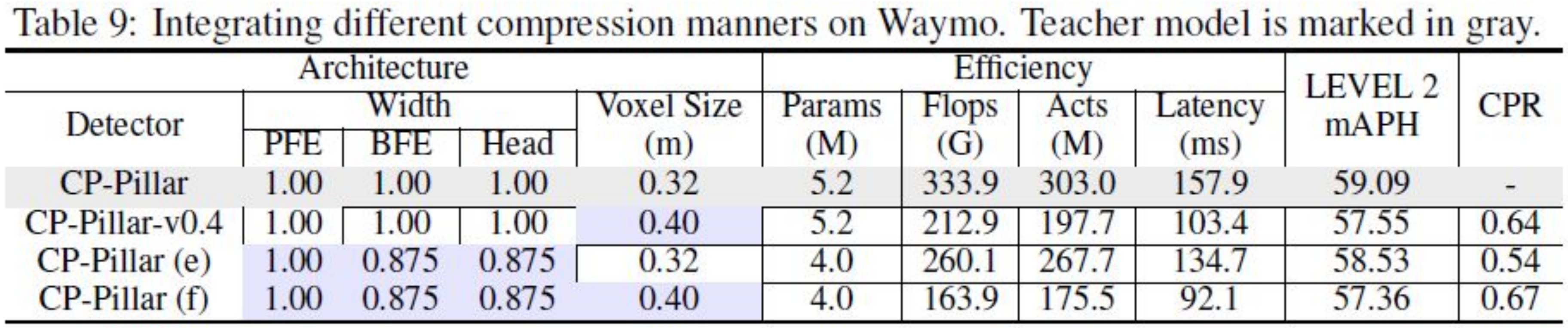


Table 9: Integrating different compression manners on Waymo. Teacher model is marked in gray.

Architecture					Efficiency					
Detector	Width			Voxel Size (m)	Params (M)	Flops (G)	Acts (M)	Latency (ms)	LEVEL 2 mAPH	CPR
	PFE	BFE	Head							
CP-Pillar	1.00	1.00	1.00	0.32	5.2	333.9	303.0	157.9	59.09	-
CP-Pillar-v0.4	1.00	1.00	1.00	0.40	5.2	212.9	197.7	103.4	57.55	0.64
CP-Pillar (e)	1.00	0.875	0.875	0.32	4.0	260.1	267.7	134.7	58.53	0.54
CP-Pillar (f)	1.00	0.875	0.875	0.40	4.0	163.9	175.5	92.1	57.36	0.67

Conclusion

- **Examined the potential of knowledge distillation to serve as a generic method for obtaining efficient 3D detectors** with extensive experimental results and analysis.
- Found that **pillar-based detector prefers input compression** while **voxel-based detector is more suitable for width compression** in designing efficient student models. Besides, we **proposed pivotal position logit KD and teacher guided initialization for enhancing the 3D KD pipeline**.
- The best performing detector outperforms its teacher with 2.4× fewer flops and the most efficient detector is 2.2× faster than previous fastest voxel/pillar-based detector PointPillars on an NVIDIA A100 with higher performance.