ICT Express 11 (2025) 696–702

Contents lists available at ScienceDirect

ICT Express

journal homepage: www.elsevier.com/locate/icte

Region-aware knowledge distillation between monocular camera-based 3D object detectors

Se-Gwon Cheon, Hyuk-Jin Shin, Seung-Hwan Bae *

Dept. of Electrical and Computer Engineering, Inha University, Incheon, Republic of Korea

ARTICLE INFO

Keywords:
3D object detection
Multi-scale feature map
Feature pyramid networks
Model compression
Knowledge distillation

ABSTRACT

Recent knowledge distillation (KD) for 3D object detection often involves costly LiDAR or multi-camera data. We focus on monocular camera-based 3D detectors, where missing 3D cues cause large feature gaps. To address this, we propose region-aware KD, aligning object features by matching their scales and pyramid levels. We introduce a probabilistic distribution to weigh region importance. Applied to MonoRCNN++ and MonoDETR on the KITTI and Waymo dataset, our approach achieves reduced complexity and strong performance with a lightweight backbone. Compared to recent KD methods, ours excels in both effectiveness and efficiency.

1. Introduction

3D object detection is to identify and localize objects in a 3D coordinate from sensor data. This task can be roughly categorized into LiDAR-based 3D detection using point clouds and camera-based 3D object detection using images. Due to the usage of depth features, the LiDAR-based detector produces more accurate results than the image-based detector. However, it is costly and constrained rather in installation. Therefore, there are many efforts to enhance the 3D accuracy of camera-based detectors. For instance, CADDN [1] utilizes a heavy backbone (ResNet-101) to generate categorical depth distribution and accurate 3D feature maps. However, these methods tend to use high computational resources due to the complexity of CNN.

To mitigate this complexity of camera-based 3D object detection, knowledge distillation (KD) methods for 3D object detection are developed. CMKD [2] performs KD to transfer the knowledge of a LiDAR model to the camera detector by aligning bird's eye view (BEV) feature maps. BEVDistill [3] also aligns BEV feature maps between a LiDAR expert and multi-camera apprentice models by focusing the distillation of the foreground region more. However, these methods require dense point clouds in a whole scene. Recently, FD3D [4] presents the KD method between multi-camera-based detectors. It performs KD on both perspective view and BEV using deformable attention. Even though it shows promising results, the extra cost of using multi-view sensory data is a burden, emphasizing the need for monocular camera-based KD. As shown in Table 1, monocular camera-based KD remains underexplored.

Consistently, a more cost-efficient KD method can transfer the knowledge of an expert detector to an apprentice detector on a monocular camera. The main bottleneck of this monocular-based KD is limited geometric features such as depth cue. As mentioned in [5], the depth is consistently associated with object 2D locality. The absence of depth features also accelerates the knowledge gaps between expert and apprentice detectors due to the higher dependency on a used feature extractor.

To minimize the knowledge gap between 3D detectors, we propose a region-aware knowledge distillation (RAKD) on monocular-camera-based 3D detection. Since object locality is the main factors to affect 3D precision usually, we transfer the knowledge to a target detector. Our region-aware KD is based on a region-of-interest (RoI) feature alignment extracted from detectors with different knowledge. To handle object geometric variation, it is necessary to employ multi-scale feature maps [6,7] and select a suitable feature-scale level of extracting an object feature. However, we observe that the one-to-one alignment between a RoI feature pair extracted at a single scale does not improve an apprentice detector dramatically. Therefore, we align a region feature pair across a whole feature scale. Since an appropriate feature scale is also relevant to object scale [7], we propose a soft assignment weight that models the geometrical relationship between feature pyramid scales and object regions as a normal distribution. As shown in Table 7, we compared normal, laplace, and uniform distributions, and observed that the normal distribution consistently achieved the highest average performance. This result led us to adopt the normal distribution for modeling the assignment weight. This is likely due to its smooth, bell-shaped curve, which provides a more balanced weighting across feature pyramid levels compared to the sharper peak of the Laplace distribution or the equal weighting of the Uniform distribution. We then use the distribution from the proposed soft assignment to compute the

* Corresponding author.
E-mail addresses: segwon1000@inha.edu (S.-G. Cheon), shin0528@inha.edu (H.-J. Shin), shbae@inha.ac.kr (S.-H. Bae).

https://doi.org/10.1016/j.icte.2025.04.012
Received 18 December 2024; Received in revised form 18 March 2025; Accepted 27 April 2025
Available online 19 May 2025

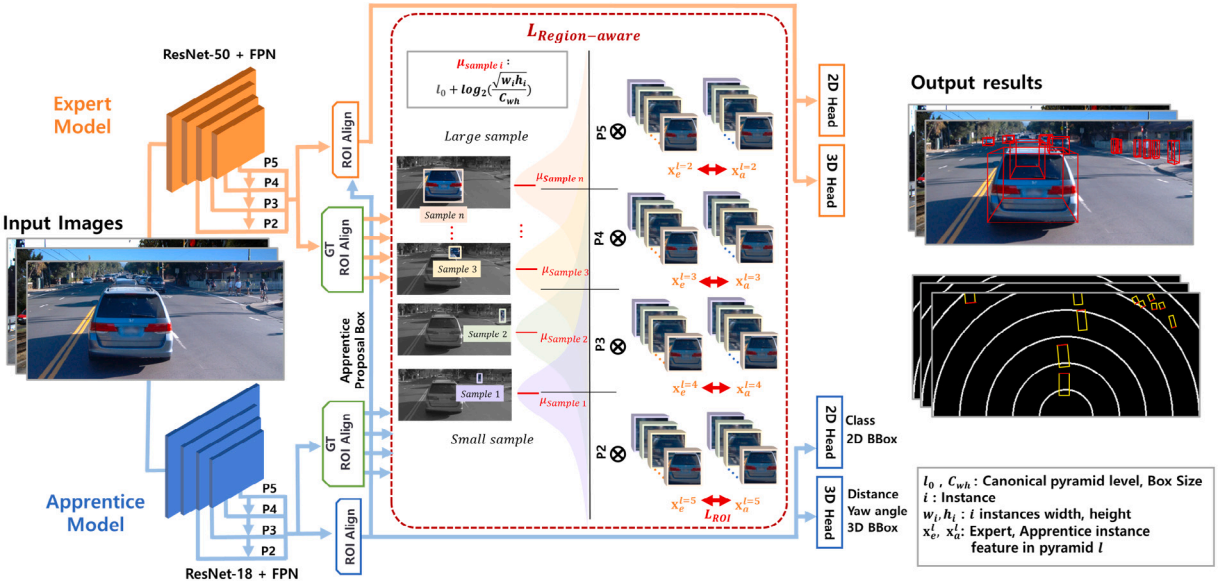

Fig. 1. The overall structure of our region-aware knowledge distillation for monocular-based 3D detection.

Table 1
Comparison sensor types of KD methods for 3D object detection: ES and AS mean expert and apprentice sensor. L, C, and MC are LiDAR, a monocular camera, and multiple cameras.

Methods	ES	AS	Additional network
CMKD [2]	L	C	–
BEVDistill [3]	L	MC	–
X3KD [15]	L	MC	Segmentation network
FD3D [4]	MC	MC	KD head
RAKD (Ours)	C	C	–

importance of a region feature per scale and use the score as coefficients of our multi-scale region feature alignment. As a result, this region-aware KD can transfer the expert knowledge of object locality and semantics across all feature scales.

To sum up, the main contributions of this work are:

- Knowledge distillation which can be applied for monocular-based 3D object detection.
- Region-aware KD to measure the geometrical relation score of object locality and feature pyramid scales and exploit it for multi-scale RoI feature alignment.
- Soft assignment weight that leverages additional pyramid levels, adapted to each object's scale, to align different semantic feature levels.

We evaluate our RAKD method by incorporating it into recent 3D object detector, MonoRCNN++ [8] and MonoDETR [9]. On the KITTI [10] and Waymo dataset [11], our apprentice detector increases detection speed by 55% compared to their experts. In addition, we compare our RAKD with other state-of-the-art KD methods: PKD [12], SemCKD [13] and GKD-BMFI [14]. Compared to PKD, SemCKD and GKD-BMFI, our method provides more detection gains by 9%, 15% and 9% in MonoRCNN++.

2. Proposed region-aware knowledge distillation method

To address the knowledge discrepancy arising from the lack of 3D geometric features in a monocular camera domain, we propose a Region-Aware Knowledge Distillation (RAKD) method between expert T_e and apprentice T_a detectors. The overall framework of the proposed RAKD is illustrated in Fig. 1. Let $\mathbf{d}^* = (x, y, w, h)$ be a bounding box, where x, y, w and h are the top-left positions, width and height. We also denote $\mathbf{d}$ a predicted bounding box from the region proposal network [16]. $\mathbf{p} \in \mathbb{R}^C$ is a predicted confidences, where C is the cardinality of object classes. Given an input image, we extract multi-scale feature maps $\{P^l\}_{l=1}^L$, $P^l \in \mathbb{R}^{H^l \times W^l \times C^l}$ from a backbone network(*e.g.* ResNet50/18 with FPN), where L is the number of feature pyramid levels, and H^l, W^l, and C^l is the height, width, and the number of channels of P^l.

We define a region-aware KD loss to transfer the knowledge T_e to T_a as follows:

$$\mathcal{L}_{\text{RAKD}} = w_{\text{RA}}\mathcal{L}_{\text{SRoI}} + w_{\text{Fit}}\mathcal{L}_{\text{Fit}} + w_{\text{Hin}}\mathcal{L}_{\text{Hin}} \tag{1}$$

where w_{RA}, w_{Fit} and w_{Hin} are balancing terms that adjust the magnitude of each loss. $\mathcal{L}_{\text{Hin}}$ is the KL divergence between $\mathbf{p}^e_{i,j}$ and $\mathbf{p}^a_{i,j}$ [17]. $\mathcal{L}_{\text{Fit}}$ is a FitNet KD loss between P^l_e and P^l_a. However, the $\mathcal{L}_{KD}$ does not transfer the intermediate feature knowledges of P^l and the object geometrical knowledge $\mathbf{d}^*$. Therefore, we present a region-aware KD to minimize the geometrical knowledge discrepancy of T_e and T_a on multi-scale features.

2.1. Region-aware feature alignment

To facilitate knowledge transfer of the object geometric locality, we focus on aligning object region features of T_e and T_a given $\mathbf{d}^*$ and $\{P^l\}_{l=1}^L$. Let $D = \{\mathbf{d}^*_i, \mathbf{p}^*_i\}_{i=1}^N$ be a set of GT bounding boxes. Along the feature pyramids $\{P^l_e\}_{l=1}^L$ and $\{P^l_a\}_{l=1}^L$ of T_e and T_a, we then can extract an object region feature $\mathbf{x}^l_{e,i}$ and $\mathbf{x}^l_{a,i}$ of the size $(h_r \times w_r \times c_r)$ at a feature scale l for $\mathbf{d}^*_i$ using RoIAlign [18], where h_r, w_r and c_r are the height, width, and the number of channels after RoIAlign. We then define the RoI feature alignment loss $\mathcal{L}_{RoI}$ as:

$$\mathcal{L}_{\text{RoI}}(\mathbf{x}^l_{e,i}, \mathbf{x}^l_{a,i}) = \frac{1}{h_r w_r}\sum_{l=1}^{L}\left|\left|\mathbf{x}^l_{e,i} - \mathbf{x}^l_{a,i}\right|\right|^2 \tag{2}$$

Eq. (2) assumes that $\mathbf{x}^l_{e,i}$ contributes equally at each scale l.

2.2. Soft feature assignment weight

Each feature at a different level contains different semantics: the lower layers contain more locality-oriented semantics, but the higher layer has high-level saliency of objectness [19]. From this insight, we additionally consider the dependency between object region and

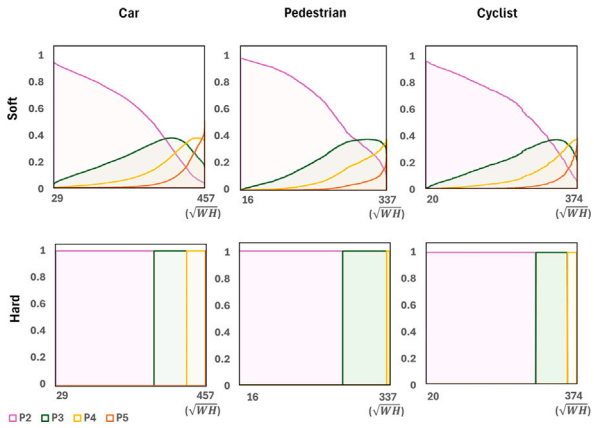

Fig. 2. The sample weights assigned based on object size ($\sqrt{WH}$) are shown for the soft assignment method (top) and the hard assignment method (bottom).

feature level. In FPN [6,20,21], the hard assignment, which matches a RoI d^* of width w and height h to the pyramid level l, is presented. Since this hard assignment does not leverage all feature scales, we present a soft weighting w_i^l by modeling the importance of each level to be a normal distribution:

$$\Phi^l(\mathbf{d}) = \int_{\rho_l}^{\rho_{l+1}} N(\mu=\mu_i,\sigma^2),\ \rho_l = \begin{cases} -\infty & \text{if } l=0 \\ \infty & \text{if } l=L+1 \\ l & \text{else} \end{cases} \\ \mu_i = l_0 + \log_2\left(\sqrt{\frac{w_i h_i}{C_{wh}}}\right) \tag{3}$$

where μ_i is the mean of the feature level for the bounding box $\mathbf{d}_i$. To evaluate this, we exploit the feature level assignment rule of FPN [6] since they design it with depth analysis of box patterns on generic object detections [22]. So, we also use the same canonical pre-training size $C_{wh} = 224$. l_0 is the base level to be tuned (in our experiment, we set it to $l_0(=4)$). $\sigma(=1)$ and $\mu(=0)$ is a hyperparameters of standard deviation. We determined these parameters with the following sensitivity experiment, as shown in Table 6. By varying σ and μ, we observed that the standard normal distribution parameters yielded the highest performance, leading us to adopt these values. Due to $\sum_{l=1}^{L} \Phi^l = 1$ of the area of probabilistic distribution, we use Φ^l as the soft-assignment weight. Subsequently, we improve L_{RoI} Eq. (2) by embedding Φ^l as:

$$\mathcal{L}_{\text{SRoI}}(\mathbf{x}_{e,i}^l, \mathbf{x}_{a,i}^l, \mathbf{d}_i^*) = \frac{1}{h_r w_r} \sum_{i=1}^{N} \sum_{l=1}^{L} \Phi^l(\mathbf{d}_i^*) \left| |\mathbf{x}_{e,i}^l - \mathbf{x}_{a,i}^l| \right|^2 \tag{4}$$

As shown in Fig. 2, we compare sample weights of a sample between hard and soft assignments given a detection $\mathbf{d}_i$. In the hard assignment, we simply determine its pyramid level by evaluating $\lfloor \mu_i \rfloor$ in Eq. (3). On the other hand, we compute $\Phi^l(\mathbf{d}_i)$ for the soft assignment. A box scales are rescaled by $\sqrt{h_i w_i}$. Horizontal and vertical axes represent a box scale and assigned weight ($[0,1]$). As can be seen, we can exploit more pyramid levels using the soft assignment since the weight of a box is distributed across scales. This is obvious benefit in multi-scale KD since different feature semantic levels can be exploited. The more comparison between the assignments can be found in Table 5.

2.3. Detection headers and losses

The total detection loss for training a detector using our RAKD of Eq. (1) can be defined as:

$$\mathcal{L}_{\text{total}} = w_{\text{RAKD}} \mathcal{L}_{\text{RAKD}} + \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{3D}}, \tag{5}$$

For $\mathcal{L}_{\text{2D}}$, we use the RPN and box classification regression losses of Faster R-CNN [16]. Followed by MonoRCNN++ [8], we compute $\mathcal{L}_{\text{3D}}$ by the predicted yaw angle and depth.

3. Experiments

We apply our RAKD method for recent monocular-based 3D detectors: MonoRCNN++ and MonoDETR detectors which designed for monocular-based camera detections.

3.1. Datasets

The KITTI dataset (7,481 training and 7,518 testing samples) is a standard 3D detection benchmark. The training set is split into 3,712 training and 3,769 validation images [8]. Objects are categorized into Easy, Moderate, and Hard, based on 2D bounding box height, occlusion, and truncation. We report AP|R40 with IoU thresholds of 0.7 for cars and 0.5 for pedestrians/cyclists.

The Waymo dataset (52,386 training and 39,848 validation samples) is a large-scale benchmark for monocular 3D detection. The training set is sampled from 798 sequences at every third frame following the CaDDN sampling protocol [1]. Objects are categorized into LEVEL 1 and LEVEL 2 based on the number of LiDAR points. Here in the with LEVEL 2, an object is assigned with five or fewer points. We report mAP across three distance ranges (0–30 m, 30–50 m, and over 50 m) using the official Waymo evaluation protocol.

3.2. Implementation details

All models are trained on two TITAN RTX 24 GB GPUs and an Intel Xeon Gold 6242 CPU. Given a pre-trained backbone (e.g., ResNet-50), we treat detectors with deeper or shallower backbones as expert or apprentice, respectively. The expert is the official pre-trained model from [8] without further tuning, while the apprentice is initialized with the same backbone weights but random FPN and heads. We apply RAKD from the expert to the apprentice. For experiments on KITTI, MonoRCNN++ is trained for 30k iterations with an initial learning rate of 0.01. The learning rate is decayed by a factor of 0.1 at 60%, 80%, and 90% of a total training iterations, and we set $w_{\text{RA}} = 1.0$. MonoDETR is trained for 135 epochs with a batch size of 8 and an initial learning rate of 0.000165. The learning rate decays at 90% epoch using the AdamW optimizer for stability. Baseline apprentices follow the same schedule without applying RAKD. On Waymo, MonoRCNN++ is trained for 30k iterations with batch of 128, an initial learning rate of 0.08, and weight decay reduced by 0.1 at 60%, 80%, and 90% of a total training iterations. We set $w_{\text{RA}} = 1.0$ same as MonoRCNN++. MonoDETR is trained for 30 epoch with batch of 40 an initial learning rate of 0.0002, and weight decay reduced by 0.001 at 18, 26 epoch of the training and set $w_{\text{RA}} = 1.0$ same as MonoRCNN++. Baseline apprentices follow the same schedule without RAKD. More precisely, during training for apprentice MonoDETR, we enforce region-aware feature alignment by generating ROIs from the ground-truth boxes and performing soft assignments using a canonical size 224 and level 5 with each instance bbox sizes. ROI pooling is then applied on pyramid features (p2–p5) from both the apprentice and teacher, and a weighted per-box MSE loss is computed.

3.3. Comparison with 3D detectors

To compare with SOTA 3D detectors, we apply RAKD to MonoRCNN++ and MonoDETR on KITTI test dataset. We Using a ResNet-50 expert and ResNet-18 apprentice, As shown in Table 2 our MonoRCNN++ and MonoDETR apprentices run in 29 ms and 40 ms, achieving 55% and 35% speedups over the experts 45 ms and 54 ms runtimes, mainly due to reduced FLOPs. For accuracy, the apprentice MonoRCNN++ further improved AP_{3D} scores cyclist classes overall when compared to the expert scores and MonoDETR further improved AP_{3D} scores pedestrian classes overall when compared to the expert scores. This improvement shows the effect of our region-aware KD which can emphasize region than other detectors [24,26]. Although the accuracy

Table 2
Comparisons with state-of-the-art 3D detectors on the KITTI test set. Expert (E) and apprentice (A) detectors applying for RAKD are shown in the last rows. Latency measured by ours on the same single NVIDIA TITAN RTX is denoted with ∗. L and GF denote usage of LiDAR and GFLOPs.

Approaches	Backbone	L	Latency	GF	Car (AP_{3D})			Pedestrian (AP_{3D})			Cyclist (AP_{3D})		
					Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Monodle [23], CVPR21	DLA-34		23ms*	158	17.23	12.26	10.29	9.64	6.55	5.44	4.59	2.66	2.45
CaDDN [1], CVPR21	ResNet-101	✓	213ms*	985	19.17	13.41	11.46	12.87	8.14	6.76	7.00	3.41	3.30
MonoEF [24], TPAMI21	DLA-34		25ms*	87	21.29	13.87	11.71	4.27	2.79	2.21	1.80	0.92	0.71
MonoJSG [25], CVPR22	DLA-34		40ms	–	24.69	16.14	13.64	11.02	7.49	6.41	5.45	3.21	2.57
MonoCon [26], CVPR22	DLA-34		26ms	–	22.50	16.46	13.95	13.10	8.41	6.94	2.80	1.92	1.55
CMKD [2], ECCV22	ResNet-50	✓	99ms	–	25.09	16.99	15.30	17.79	11.69	10.09	9.60	5.24	4.50
DEVIANT [27] , ECCV22	DLA-34		75ms*	535	21.88	14.46	11.89	13.43	8.65	7.69	5.05	3.13	2.59
MonoRCNN++ (E), WACV23	ResNet-50		45ms*	216	20.08	13.72	11.34	12.26	7.90	6.62	3.17	1.81	1.75
MonoRCNN++ w/t RAKD (A)	ResNet-18		29ms*	138	18.45	11.71	9.32	12.25	7.99	6.66	3.61	2.16	1.76
MonoDETR (E), ICCV23	ResNet-50		54ms*	108	22.78	14.97	12.23	13.50	8.64	7.12	6.19	3.91	3.20
MonoDETR w/t RAKD (A)	ResNet-18		40ms*	71	19.23	12.57	10.07	13.74	8.83	7.31	3.44	2.09	1.75

Table 3
Detailed comparisons of detectors with/without our RAKD on the Waymo val set. We use 3D AP ($IoU > 0.5$) as a standard metric for all classes. All the scores of the detectors are evaluated by our reimplementation.

Difficulty	Approaches (Backbone)	Expert/Apprentice	Vehicle AP_{3D} [%] ↑				Pedestrian AP_{3D} [%] ↑			Cyclist AP_{3D} [%] ↑		
			Overall	0–30	30–50	50-∞	Overall	0–30	30–50	Overall	0–30	30-50
LEVEL 1	MonoRCNN++ [ResNet-50]	Expert	11.05	25.90	4.23	0.89	5.40	13.92	1.74	3.79	8.37	0.10
	MonoRCNN++ [ResNet-18]	Apprentice	9.12	23.01	2.91	0.64	4.34	11.51	1.24	0.77	2.22	0.00
	MonoRCNN++ [ResNet-18] + RAKD	Apprentice	9.15	22.39	3.43	0.57	5.29	13.24	1.86	1.27	3.69	0.00
	MonoDETR [ResNet-50]	Expert	9.04	24.92	3.84	0.31	5.73	15.96	2.43	5.75	15.75	0.55
	MonoDETR [ResNet-18]	Apprentice	7.81	22.06	3.05	0.28	4.90	13.97	1.84	4.92	12.81	0.87
	MonoDETR [ResNet-18] + RAKD	Apprentice	8.02	21.81	3.46	0.33	4.92	14.38	1.63	5.07	14.16	0.22
LEVEL 2	MonoRCNN++ [ResNet-50]	Expert	10.31	25.79	4.08	0.77	4.91	13.70	1.63	3.65	8.32	0.09
	MonoRCNN++ [ResNet-18]	Apprentice	8.50	22.92	2.80	0.55	3.95	11.33	1.16	0.74	2.21	0.00
	MonoRCNN++ [ResNet-18] + RAKD	Apprentice	8.52	22.30	3.43	0.57	4.81	13.03	1.74	1.23	3.68	0.00
	MonoDETR [ResNet-50]	Expert	8.47	24.83	3.71	0.27	5.21	15.72	2.28	5.53	15.67	0.52
	MonoDETR [ResNet-18]	Apprentice	7.32	21.98	2.94	0.24	4.46	13.76	1.72	4.73	12.75	0.82
	MonoDETR [ResNet-18] + RAKD	Apprentice	7.51	21.73	3.34	0.28	4.48	14.16	1.52	4.88	14.09	0.21

Table 4
Comparisons with the recent PKD, SemCKD and GKD-BMFI methods on the KITTI validation set. The best results and second-best scores are highlighted in bold and underlined (except for expert). E and A also denote expert and apprentice detectors. All the latency is measured on the same machine. Avg. represents the score averaged for all classes and occlusion levels.

3D Detector	Backbone	Methods	Car (AP_{3D})			Pedestrian (AP_{3D})			Cyclist (AP_{3D})			Avg.	Latency
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard		
MonoRCNN++	ResNet-50 (E)	Vanilla	19.13	14.69	12.42	6.85	5.58	4.58	4.38	2.48	2.57	8.08	45ms
	ResNet-18 (A)	Vanilla	17.71	13.19	10.43	7.05	5.09	3.94	2.70	1.52	1.29	6.99	29ms
		PKD [12]	19.66	13.40	10.98	6.65	4.81	4.29	**3.78**	**2.11**	**2.19**	7.54	29ms
		SemCKD [13]	17.96	12.63	10.39	8.03	5.97	4.66	2.38	1.33	1.26	7.18	29ms
		GKD-BMFI [13]	18.79	12.92	10.56	9.41	6.68	5.26	1.78	1.46	1.14	7.56	29ms
		RAKD (Ours)	**20.75**	**14.45**	**12.02**	**8.56**	**6.19**	**5.02**	3.59	1.78	1.81	**8.24**	29ms
MonoDETR	ResNet-50 (E)	Vanilla	27.69	19.53	16.22	10.05	7.06	5.51	8.79	4.46	4.06	11.49	54ms
	ResNet-18 (A)	Vanilla	22.24	16.23	13.48	9.34	6.82	5.49	6.54	3.48	3.47	9.68	40ms
		PKD	25.06	**17.95**	**14.94**	9.71	**7.33**	5.70	**9.03**	4.52	**4.38**	**10.96**	40ms
		SemCKD	24.14	16.67	13.82	8.86	5.95	5.22	8.68	4.49	4.29	10.24	40ms
		GKD-BMFI	**25.14**	17.12	14.11	7.78	5.90	4.85	5.77	2.48	2.51	9.52	40ms
		RAKD (Ours)	24.19	16.79	13.77	**10.51**	7.32	**5.87**	8.75	**4.55**	4.15	10.66	40ms

of our apprentice detectors is somewhat lower than other recent SOTA detectors [2,27], our detectors achieve competitive accuracy with much lower latency and FLOPs. These comparison results clearly show the effect of our RAKD on recent 3D detectors. To further evaluate the effectiveness of RAKD, we apply it to MonoRCNN++ and MonoDETR on the Waymo validation dataset. As shown in Table 3, applying RAKD to MonoRCNN++ results in consistently higher accuracy across all classes compared to the apprentice model. Similarly, for MonoDETR, RAKD improves performance over the apprentice model in all classes except for pedestrians. These results demonstrate that RAKD effectively enhances 3D detection accuracy across different architectures, reinforcing its ability to transfer knowledge efficiently while maintaining competitive performance.

3.4. Comparison of knowledge distillation

In addition, our RAKD methods are compared with other recent KD methods on the KITTI validation set as shown in Table 4. We adopt PKD [12], SemCKD [13] and GKD-BMFI [14] since they can distill multi-scale intermediate features. For applying the SemCKD method for those detectors, we transform each stage feature output (P_e^l and P_a^l) to a tensor with the same dimensionality of T_a and T_e by using projection and reshape operations. We then evaluate the self-similarity of each stage output tensor by using dot product and MLP operations. Then, we apply a soft-max function for the MLP outputs at layer l of T_a and T_e to generate attention-based weights. We use this attention as a layer-wise weight when evaluating $\mathcal{L}_{\text{Fit}}$ of Eq. (1). In addition, for implementing PKD, we reshape P_e^l and P_a^l of size $h_l \times w_l \times c_l$ to

Table 5
Ablation study on the KITTI validation set. $\mathcal{L}_{RoI}$ represents RoI feature alignment loss, HFA represents hard feature assignment and SFA represents soft feature assignment ($\mathcal{L}_{SRoI}$). The best results are highlighted in bold, and the second-best results are underlined (except for expert).

Approaches	Methods			Car (AP_{3D})			Pedestrian (AP_{3D})			Cyclist (AP_{3D})			avg.
	L_{ROI}	HFA	SFA	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	
ResNet-50 (E)				19.13	14.69	12.42	6.85	5.58	4.58	4.38	2.48	2.57	8.08
ResNet-18 (A)				17.71	13.19	10.43	7.05	5.09	3.94	2.70	1.52	1.29	6.99
RAKD	✓			19.99	14.34	11.91	7.68	5.69	4.53	3.78	1.97	1.94	7.98
	✓	✓		18.80	13.64	11.47	9.13	6.81	5.25	3.39	1.83	1.80	8.01
	✓		✓	20.75	**14.45**	**12.02**	8.56	6.19	5.02	3.59	1.78	1.81	8.24
Adaptive alignment	–	–	–	18.28	13.58	11.42	7.29	5.47	4.22	4.89	**2.85**	**2.81**	7.87
PKD				19.66	13.40	10.98	6.65	4.81	4.29	3.78	2.11	2.19	7.54
PKD w/t RAKD	✓		✓	**20.81**	14.05	11.65	8.07	6.03	4.76	4.03	2.03	1.99	8.16
SemCKD				17.96	12.63	10.39	8.03	5.97	4.66	2.38	1.33	1.26	7.18
SemCKD w/t RAKD	✓		✓	19.37	13.91	11.59	8.71	6.18	4.91	4.60	2.36	2.25	8.21
GKD-BMFI				18.79	12.92	10.56	9.41	6.68	5.26	1.78	1.46	1.14	7.56
GKD-BMFI w/t RAKD	✓		✓	19.69	13.83	11.41	**9.99**	**7.05**	**5.59**	**4.94**	2.82	2.37	**8.63**

Table 6
Comparisons with the sensitive parameter to normal distribution on the KITTI validation set. The best results scores are highlighted in bold. Avg. represents the score averaged for all classes and occlusion levels.

Method	Parameter	Value	Car (AP_{3D})			Pedestrian (AP_{3D})			Cyclist (AP_{3D})			Avg.
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	
RAKD	Vanilla ($\mu = 0$, $\sigma = 1$)		**20.75**	**14.45**	12.02	**8.56**	6.19	**5.02**	3.59	1.78	1.81	**8.24**
	μ	−0.4	18.79	13.00	10.73	6.80	4.31	4.07	4.46	2.15	2.22	7.39
		−0.2	18.02	13.18	11.26	8.37	**6.26**	4.95	3.36	1.79	1.83	8.07
		+0.2	19.20	14.19	**12.03**	7.60	5.33	4.61	**4.99**	**2.73**	**2.67**	8.15
		+0.4	19.30	14.38	12.02	7.57	5.97	4.61	**4.99**	2.16	2.21	7.66
	σ	0.5	18.29	13.42	11.30	7.89	6.00	4.71	2.74	1.42	1.26	7.44
		2.0	18.31	13.59	11.38	8.54	6.22	4.76	4.04	2.20	2.02	7.89

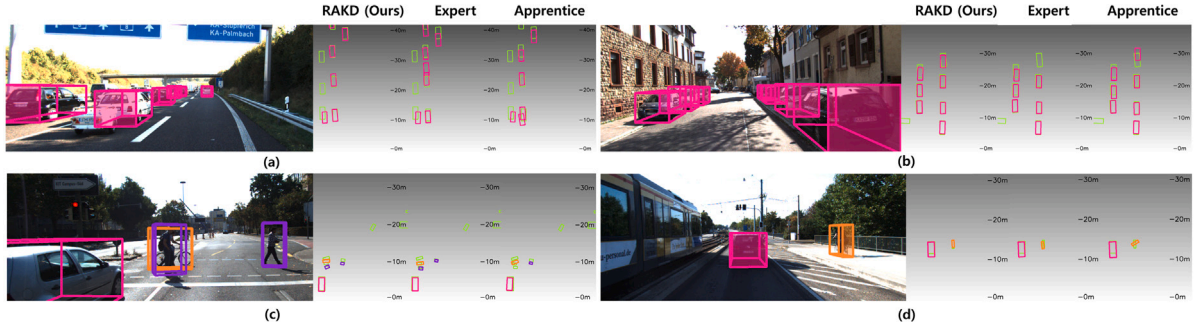

Fig. 3. Qualitative comparison between expert and apprentice MonoRCNN++ detectors with our RAKD. The left image represents the 3D box coordinates projected onto a 2D plane image, and the right image shows the position of the 3D object in a BEV. The green, pink, purple and orange lines show the GT and predicted car, pedestrian and cyclist detection results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$c_l \times h_l \cdot w_l$. Then, we compute the Pearson correlation coefficient between the reshaped vectors with the mean and covariance of each vector. Then, we perform FPN distillation with the normalized coefficients. To implement GKD-BMFI, the feature outputs (P_e^l and P_a^l) are utilized to ensure that the multi-scale intermediate features are properly aligned. Subsequently, two sets of CAM masks, one from the expert network and the other from the apprentice, are obtained by averaging the feature maps across channels c_l and applying a sigmoid function. Following the methodology outlined in the GKD-BMFI paper, a Gaussian mask is generated based on the bounding box $\mathbf{d}^*$ with w_l and h_l. Table 4 shows the comparison results of KD methods. We also present the accuracy scores of each vanilla apprentice detector. All the KD methods achieve more precision gains than the vanilla detector. In most metrics, our RAKD shows the best precision rates. In particular, In MonoRCNN++ our RAKD is very effective for improving more difficult class detection such as Car in Hard difficulty and Pedestrians. For MonoDETR our RAKD significantly boosts pedestrian detection, resulting in a balanced overall performance. These results show the superiority of our region-aware KD method.

3.5. Ablation study

Table 5 shows the effect of each method. For this ablation, we apply our method one by one. Since $\mathcal{L}_{SRoI}$ of Eq. (4) consists of $\mathcal{L}_{RoI}$ of Eq. (2) and (3), we evaluate its effect separately. We also compare the hard-level feature assignment (HFA) of [6] by using μ_i of Eq. (3). In the ablation study of RAKD, we know that the hard assignment does not work well for 3D KD. On the other hand, our RAKD provides more gain than HFA.

We also compare our RAKD with learnable assignments (Adaptive alignment) paradigm. To implement this, we modified the network so that the FPN assignment levels are produced as logits. Initially, all weights for the FPN levels were set to $1/L$, and then they were learned by minimizing the loss. The experimental results for this adaptive alignment are presented in Table 5 as adaptive alignment. As shown, the performance obtained using these learnable assignments exhibits lower accuracy than that achieved with our normal distribution–based alignment. In addition, we apply this ablation study to other KD methods to investigate the flexibility of our method. We can improve the

Table 7
Comparison of RAKD performance using different probabilistic models. Normal, Laplace, Uniform denote Normal distribution, Laplace distribution, Uniform distribution.

Method	Prob. Model	Car (AP_{3D})			Pedestrian (AP_{3D})			Cyclist (AP_{3D})			Avg.
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	
RAKD	Normal	**20.75**	**14.45**	**12.02**	8.56	6.19	5.02	3.59	1.78	1.81	**8.24**
	Laplace	17.03	13.07	10.32	7.79	5.55	4.60	**4.68**	**2.31**	**1.96**	7.49
	Uniform	18.80	13.64	11.47	**9.13**	**6.81**	**5.25**	3.39	1.83	1.80	8.01

AP scores of PKD, SemCKD and GKD-BMFI by 8.22%, 14.34% and 14.15% on average over all classes and levels. These results show that our RAKD has high compatibility with other KD methods. Furthermore, we conduct a sensitive analysis by changing the parameters of the normal distribution. From this results, we confirmed that the normal distribution with $\sigma = 1$, $\mu = 0$ achieves the best gains, and fit them for other RAKD evaluation.

Also, we compared performance with normal distribution, laplace distribution, and uniform distribution in Table 7 to confirm that normal distribution is most accurate and most suitable for the experiment. The results confirm that the normal distribution achieves the highest accuracy and is the most suitable choice for this experiment. This is because the normal distribution provides a smooth and balanced weighting across feature pyramid levels, effectively capturing gradual variations in object scales, whereas the Laplace distribution tends to overemphasize central values, and the uniform distribution is inefficient with equal weight allocation.

3.6. Qualitative comparison

Fig. 3 visualizes both the 2D projections of 3D bounding boxes and their BEV representations. Our RAKD method outperforms Expert and Apprentice models in detecting distant cars (a, b) due to the robust SROI method, which handles geometric variations. As shown in (c), it also detects overlapping pedestrians and cyclists, showcasing its robustness with occluded objects. Additionally, as shown in (d), the accurate detection of distant cyclists highlights the effective of our knowledge distillation process.

4. Conclusion

Despite advancements in knowledge distillation [2,28–30], monocular-based 3D detection remains underexplored. To address this, we propose a novel region-aware KD (RAKD) method that leverages multi-scale region features with a soft weighting mechanism and soft RoI feature alignment loss for precise KD. Using RAKD, we implement on MonoRCNN++ and MonoDETR to demonstrate significant improvements in precision and latency compared to SOTA 3D detectors. Our method also outperforms recent KD approaches and enhances their performance when integrated. We establish RAKD as a strong baseline for monocular-based 3D KD, offering efficient and accurate 3D object detection.

CRediT authorship contribution statement

Se-Gwon Cheon: Writing – original draft, Visualization, Validation, Software. **Hyuk-Jin Shin:** Writing – review & editing, Formal analysis, Data curation. **Seung-Hwan Bae:** Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by INHA UNIVERSITY Research Grant.

References

[1] C. Reading, A. Harakeh, J. Chae, S.L. Waslander, Categorical depth distribution network for monocular 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8555–8564.
[2] Y. Hong, H. Dai, Y. Ding, Cross-modality knowledge distillation network for monocular 3d object detection, in: European Conference on Computer Vision, Springer, 2022, pp. 87–104.
[3] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, F. Zhao, BEVDistill: Cross-modal BEV distillation for multi-view 3D object detection, in: The Eleventh International Conference on Learning Representations, 2023.
[4] J. Zeng, L. Chen, H. Deng, L. Lu, J. Yan, Y. Qiao, H. Li, Distilling focal knowledge from imperfect expert for 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 992–1001.
[5] Q. Lian, B. Ye, R. Xu, W. Yao, T. Zhang, Exploring geometric consistency for monocular 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1685–1694.
[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[7] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.
[8] X. Shi, Z. Chen, T.-K. Kim, Multivariate probabilistic monocular 3D object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 4281–4290.
[9] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, P. Gao, Monodetr: Depth-guided transformer for monocular 3d object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9155–9166.
[10] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.
[11] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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, 2020, pp. 2446–2454.
[12] W. Cao, Y. Zhang, J. Gao, A. Cheng, K. Cheng, J. Cheng, Pkd: General distillation framework for object detectors via pearson correlation coefficient, Adv. Neural Inf. Process. Syst. 35 (2022) 15394–15406.
[13] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, C. Chen, Cross-layer distillation with semantic calibration, in: Proceedings of the AAAI Conference on Artificial Intelligence, 35, (8) 2021, pp. 7028–7036.
[14] Q. Lan, Q. Tian, Gradient-guided knowledge distillation for object detectors, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 424–433.
[15] M. Klingner, S. Borse, V.R. Kumar, B. Rezaei, V. Narayanan, S. Yogamani, F. Porikli, X3kd: Knowledge distillation across modalities, tasks and stages for multi-camera 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13343–13353.
[16] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, NIPS 28 (2015).
[17] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint arXiv:1503.02531.
[18] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[19] Y. Pang, Y. Li, J. Shen, L. Shao, Towards bridging semantic gap to improve semantic segmentation, in: ICCV, 2019, pp. 4230–4239.
[20] S.-H. Lee, S.-H. Bae, AFI-gan: Improving feature interpolation of feature pyramid networks via adversarial training for object detection, Pattern Recognit. 138 (2023) 109365.
[21] Y. Jung, N.S. Syazwany, S. Kim, S.-C. Lee, Fine-grained classification via hierarchical feature covariance attention module, IEEE Access 11 (2023) 35670–35679.

[22] S.-H. Bae, Deformable part region learning and feature aggregation tree representation for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 10817–10834.
[23] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, W. Ouyang, Delving into localization errors for monocular 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4721–4730.
[24] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, Q. Jiang, Monoef: Extrinsic parameter free monocular 3D object detection, IEEE Trans. Pattern Anal. Mach. Intell. 44 (12) (2021) 10114–10128.
[25] Q. Lian, P. Li, X. Chen, Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1070–1079.
[26] X. Liu, N. Xue, T. Wu, Learning auxiliary monocular contexts helps monocular 3d object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, 36, (2) 2022, pp. 1810–1818.
[27] A. Kumar, G. Brazil, E. Corona, A. Parchami, X. Liu, Deviant: Depth equivariant network for monocular 3d object detection, in: European Conference on Computer Vision, Springer, 2022, pp. 664–683.
[28] H.-B. Bak, S.-H. Bae, Bridging the knowledge gap via transformer-based multi-layer correlation learning, IEEE Access (2024).
[29] M. Yoon, S. Lee, B.C. Song, TAKDSR: Teacher assistant knowledge distillation framework for graphics image super-resolution, IEEE Access 11 (2023) 112015–112026.
[30] S. Lee, B.C. Song, Fast filter pruning via coarse-to-fine neural architecture search and contrastive knowledge transfer, IEEE Trans. Neural Networks Learn. Syst. (2023).