ECAI 2023
K. Gal et al. (Eds.)

© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230555

MonoSKD: General Distillation Framework for Monocular 3D Object Detection via Spearman Correlation Coefficient

Sen Wang[a] **and Jin Zheng** [a;*]

[a]School of Computer Science and Engineering, Beihang University, Beijing, China, 100191

Abstract. Monocular 3D object detection is an inherently ill-posed problem, as it is challenging to predict accurate 3D localization from a single image. Existing monocular 3D detection knowledge distillation methods usually project the LiDAR onto the image plane and train the teacher network accordingly. Transferring LiDAR-based model knowledge to RGB-based models is more complex, so a general distillation strategy is needed. To alleviate cross-modal problem, we propose **MonoSKD**, a novel **K**nowledge **D**istillation framework for **Mono**cular 3D detection based on **S**pearman correlation coefficient, to learn the relative correlation between cross-modal features. Considering the large gap between these features, strict alignment of features may mislead the training, so we propose a looser Spearman loss. Furthermore, by selecting appropriate distillation locations and removing redundant modules, our scheme saves more GPU resources and trains faster than existing methods. Extensive experiments are performed to verify the effectiveness of our framework on the challenging KITTI 3D object detection benchmark. Our method achieves state-of-the-art performance until submission with no additional inference computational cost. Our codes are available at https://github.com/Senwang98/MonoSKD.

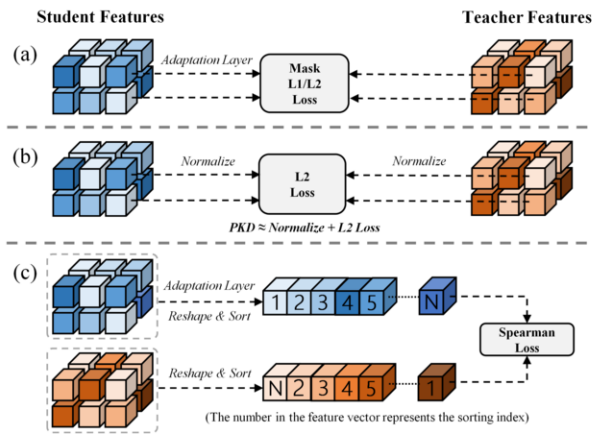

Figure 1: **Comparison of different distillation strategies.** (a) The loss of mask L1 used in MonoDistill [9], where the mask is used to filter the background. (b) The Pearson loss proposed in PKD [4] is equivalent to normalization combined with L2 loss. (c) Our Spearman distillation loss for monocular 3D detection.

1 Introduction

Due to its widespread applications, 3D object detection has attracted significant attention in augmented reality, autonomous driving, and robot navigation. Accurate 3D localization is the basis for ensuring security, so the key to 3D object detection is to obtain accurate 3D localization. According to the input resources, the existing 3D object detectors can be divided into four categories: LiDAR point cloud-based [19], stereo image-based [13], monocular image-based [31, 42] and multi-modality-based methods [36]. The industry usually chooses LiDAR point cloud-based, stereo image-based, or multi-modality-based methods because they can directly perceive the depth information of surroundings. Compared with LiDAR sensors and stereo cameras, monocular cameras are low-cost and flexible for deployment. Considering the above unique advantages, the community has begun to pay more attention to monocular 3D object detection.

Notable progress in monocular 3D object detection has been achieved in recent years. Nevertheless, there still exists a considerable performance gap between the monocular image-based and LiDAR-based methods. Compared with the direct acquisition of accurate depth by LiDAR, predicting depth from monocular images is an inherently ill-posed problem [23]. To mitigate this issue, several works [23] take monocular depth estimation networks to provide dense depth maps. Recently, some works [18, 41] view depth estimation as an auxiliary task to introduce depth-aware features for object detectors, achieving remarkable performance improvement. However, these detectors are still not robust enough for depth estimation, resulting in inevitable depth estimation errors.

To alleviate the above problem, MonoDistill [9] proposes a monocular 3D detection knowledge distillation (KD) framework to transfer the depth-related knowledge of LiDAR signals to the RGB student model to improve the robustness of the detector. MonoDistill projects the LiDAR signals into the image plane to generate the depth maps and trains the teacher model. In this case, directly forcing the teacher and student to align at pixel level on the feature maps is sub-optimal because cross-modal differences will mislead training. Here, LiDAR signals are only used for training, and the student network is a standard monocular 3D detection network during the inference stage. Moreover, PKD [4] has proven that applying normalization in 2D object detection can bridge the feature gap between the student and the teacher. However, due to vast modal differences, PKD only

* Corresponding Author. Email: JinZheng@buaa.edu.cn

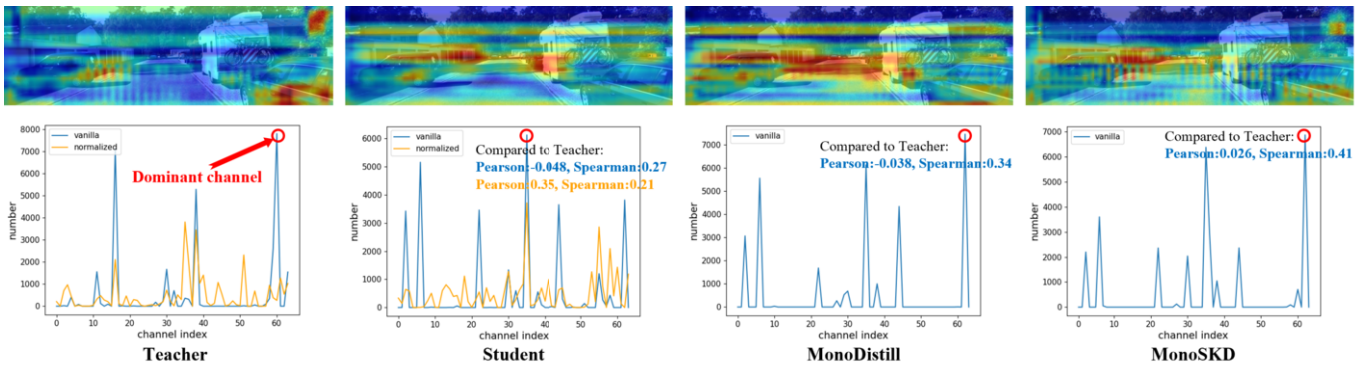

Figure 2: **Visualization of the feature maps and dominant channels. Top**: Visualization of teacher and student's neck feature maps. **Bottom**: Dominant channels in neck stage 'P2'. Let $s_{l,u,v} \in \mathbb{R}^C$ denote the feature vector of pixel (u, v) from l-th neck stage and omit l for clarity. Then $number_i = \sum_{u,v} \mathbb{1}[\text{argmax}_c\, s^{(c)}_{u,v} = i]$ where i denotes the channel index.

has a slight improvement in the cross-modal task. In Figure 1 (a) and (b), we show the two feature distillation strategies adopted by MonoDistill and PKD. The Mask L1 loss used in MonoDistill directly aligns the foreground regions of the feature maps from teacher and student, and PKD aligns normalized feature maps. At the top of Figure 2, the visualized feature maps show huge differences between teacher and student, especially in foreground regions. In contrast, the visual feature map of MonoSKD is closer to that of the teacher. The blue curve at the bottom of Figure 2 represents the dominant channel of the naive feature maps. The teacher's and student's dominant channels show the vast channel difference. Meanwhile, we migrate the PKD distillation strategy to the monocular 3D detection field and use the yellow curve to represent the dominant channel of the normalized feature maps. Compared to the student, MonoDistill and MonoSKD are more similar to teachers on the blue curve. For quantitative evaluation, we choose Pearson and Spearman correlation coefficients as quantitative metrics and compare student, MonoDistill, and MonoSKD with the teacher's dominant channel curve. The closer the metrics are to 1, the better. Even though the feature difference after normalization is reduced (-0.048 $\rightarrow$ 0.35), it still faces vast channel differences. Normalization will destroy the dominant channel ranking relationship, so the Spearman correlation coefficient drops (0.27 $\rightarrow$ 0.21). Therefore, although PKD has improved on the Pearson metric, it can disrupt the ranking relationship between features, thereby reducing the Spearman metric. Compared with MonoDistill, the MonoSKD proposed in this paper is more similar to the teacher's dominant channel. MonoSKD achieves better results than MonoDistill on both metrics because feature maps of different modalities have significant pixel-level differences, making it challenging to satisfy strict pixel alignment.

Considering the above deficiencies, we try to mine more general knowledge of cross-modal features, such as relative relationships. Therefore, we introduce the Spearman correlation coefficient (SCC) in the monocular 3D detection distillation framework to learn the ranking relationship between cross-modal features. In Figure 1 (c), we show the process of Spearman loss. It can be seen that Spearman loss only cares about sorting information between features rather than specific values, which is more suitable for cross-modal tasks.

Besides, we find that the existing distillation framework suffers from redundant distillation modules. We select the appropriate distillation location and remove redundant modules, which saves about 30% of the average GPU memory usage, accelerates training, and improves distillation performance.

To verify our scheme's effectiveness and generality, we perform distillation experiments on three recent monocular 3D detectors, including MonoDLE [24], GUPNet [22], and DID-M3D [29]. As expected, our method dramatically improves the performance of these three detectors on the KITTI [11] benchmark.

In summary, our contributions are listed as follows:

- We propose a general Spearman distillation strategy for the knowledge distillation task of monocular 3D detection to learn the ranking relationship between features and improve performance.
- We find that MonoDistill suffers from redundant distillation modules, and our redesigned distillation framework saves an average of 30% of GPU memory and accelerates training while improving distillation performance.
- We conduct extensive experiments on three detectors using the challenging KITTI benchmark to demonstrate the effectiveness and generality of our framework. Our method achieves state-of-the-art performance with no extra inference computational cost.

2 Related Work

2.1 Monocular 3D object detection

Given an input image, monocular 3D object detection aims to predict a 3D bounding box represented by its location, dimension, and orientation for each object. Based on whether to use additional data, existing methods can be divided into two categories: standard monocular 3D detectors and detectors using additional data. Standard monocular 3D detection methods such as MonoDLE only use the RGB images, annotations, and camera calibrations provided by KITTI dataset [11] to predict 3D bounding boxes. Mousavian et al. [26] combined estimated 3D object orientation and dimensions with the geometric constraints on translation imposed by the 2D bounding box to recover 3D locations. MonoPair [8] encoded spatial constraints for partially-occluded objects from their adjacent neighbors to improve the monocular 3D object detection. Qin et al. proposed MonoGRNet [30] for monocular 3D object detection via geometric reasoning in both the observed 2D projection and the unobserved depth dimension. OFTNet [32] introduced an orthographic feature transform to map image-based features into an orthographic 3D space. Instead of directly regressing depth, some works such as GUPNet predicted 2D and 3D heights via uncertainty modeling and recovered 3D locations

based on geometric priors. Meanwhile, several works [20, 42] utilized keypoint-based geometric constraints to improve the monocular 3D object detection further. These standard monocular 3D object detectors had achieved prominent progress but still suffered from the unsatisfactory performance of the 3D locations.

On the other hand, some methods use additional data. Deep Manta [5] improved performance by using more detailed annotated locations of key points, e.g., wheels, as training labels. several works [25] use the CAD models as shape templates to get better object geometry. Specifically, AutoShape [21] generated shape-aware key points via CAD models to boost the detection performance. Several works [10, 23] accomplished monocular 3D object detectors by directly taking depth maps from off-the-shelf depth estimators as extra inputs. Other works utilize additional data to help with online depth estimation. For example, DD3D [27] utilized a large private dataset and the KITTI depth dataset for depth pre-training to improve detection performance. MonoDTR [18] proposed an end-to-end transformer for monocular 3D object detection, which utilized depth maps as auxiliary supervision. Recently, DID-M3D introduced dense depth maps to decouple the instance depth. However, these detectors are still not robust enough due to unavoidable depth errors.

2.2 Knowledge distillation

The concept of knowledge distillation (KD) [15] was first proposed for model compression, which trains student models with GT labels and soft labels from teacher networks. Instead of transferring knowledge from teachers' responses, Romero et al. [33] proved that intermediate features distillation can also guide the training of student networks. After that, more and more tasks utilized knowledge distillation to achieve a remarkable performance improvement, such as object detection [4] and semantic segmentation [35], etc. Specifically, KD in object detection is usually divided into three categories: feature-based, relation-based, and response-based. Feature-based KD usually carefully designs a mask for knowledge localization and transfers knowledge after feature alignment (such as dimension and semantic alignment). Relation-based KD considers the difference in feature relations instead of the pixel-to-pixel difference between corresponding feature maps. Response-based KD is a commonly used and efficient distillation method, using the teacher's prediction as a soft label to supervise the student network.

For object detection, Chen et al. [6] first introduced knowledge distillation to 2D object detection, which distilled the neck, regression head, and classification head of detectors. Afterward, Wang et.al [38] proposed a fine-grained feature imitation method. Instead of distilling foreground object regions, Guo et al. [12] proposed that the distillation of background regions can also be effective. Recently, PKD introduced Pearson correlation coefficient (PCC) [1] and normalization strategy for homogeneous and heterogeneous detectors. For 3D object detection, Guo et al. proposed LiGA-Stereo [13] to learn stereo-based 3D detectors under the guidance of high-level geometry-aware representations of LiDAR-based detection models. PointDistiller [40] designed a structured knowledge distillation framework for point clouds-based 3D detection. MonoDistill achieved state-of-the-art performance by proposing a teacher model based on inputs of a projected LiDAR signals to guide student detectors with spatial cues for monocular 3D object detection. Nevertheless, MonoDistill's strict alignment of cross-modal features is suboptimal.

3 Methodology

3.1 Overview and Framework

General monocular 3D detection takes an image captured by an RGB camera as input, predicting 3D bounding boxes of objects for each object in 3D space. The 3D bounding boxes are usually represented by 3D center location (x, y, z), dimension (h, w, l), and the orientation θ. Most existing monocular 3D detectors obtain 2D features through the backbone and neck and recover 3D locations through multiple independent heads.

As shown in Figure 3, our distillation method differs from the existing MonoDistill in the knowledge distillation loss function and the selection of distillation location. On the one hand, we abandon the strict feature alignment distillation strategy and introduce a relation-based Spearman distillation loss, which adopts a more general distillation alignment strategy and is more suitable for cross-modal distillation. On the other hand, MonoDistill fuses and distills the output features of the backbone. We find that MonoDitill has redundant distillation modules, resulting in inefficient distillation. Thus, we use the neck output feature as the distillation object, so the heavy feature fusion module is removed, which allows us to improve performance while saving about 30% of GPU memory and speeding up training.

3.2 Knowledge Distillation with Spearman Correlation Coefficient

We empirically find that the discrepancy of predictions between the RGB-based student and LiDAR-based teacher may tend to be pretty severe. In this case, directly forcing the teacher and student to align at pixel level on the feature maps is suboptimal because cross-modal differences will mislead training. Instead of strictly aligning features, we guide distillation training with looser constraints. Recently, PKD introduced the Pearson correlation coefficient for object detection. PKD has proved that applying PCC for feature maps is equivalent to normalization combined with mean square error. The normalization mechanism bridges the gap between the activation patterns of the student and the teacher. However, our experiment shows a small gain when applying PKD to monocular 3D detection distillation (see Appendix A.6). As is shown in Figure 2, the activation patterns of the teacher and student networks are seriously different. Although normalization alleviates the differences between teachers and students, the differences in the cross-modal task are still considerable.

Considering the enormous cross-modal differences, the direct alignment of specific values between feature maps is too rigorous for network training. To seek a looser distillation strategy, we consider the relative correlation between distillation features, so the Spearman correlation coefficient is introduced. Such a distillation strategy can reduce the difficulty of distillation and further improve performance. We adopt Spearman's distance as the metric, $i.e.$,

$$\mathcal{L}_{scc}(s,t) = \frac{1}{L}\sum_{l=1}^{L}(1 - r_{scc}(s_l, t_l)) \tag{1}$$

r_{scc}, L, s, t represent the Spearman correlation coefficient, number of the feature maps from neck, student feature maps, and teacher

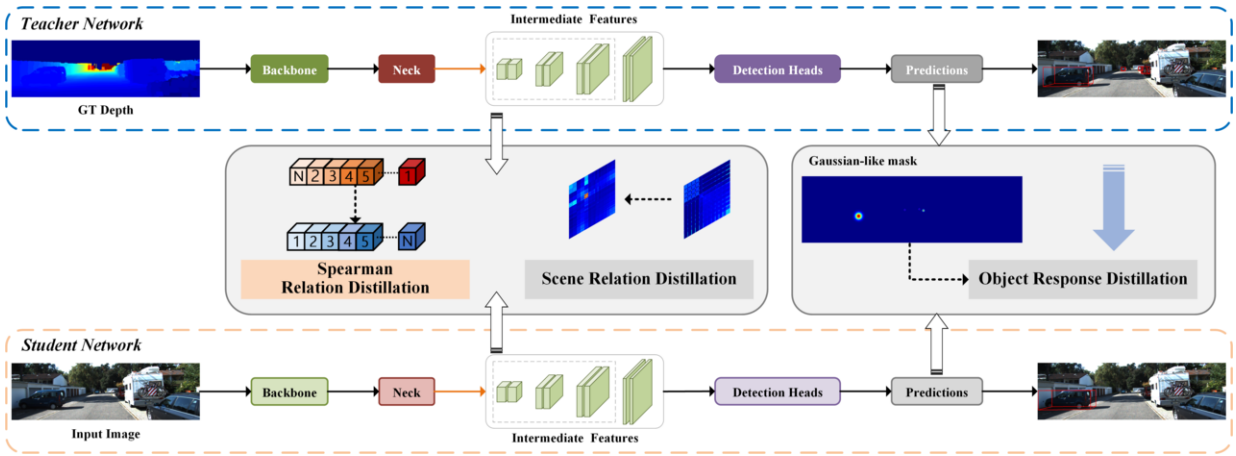

Figure 3: **Illustration of the proposed MonoSKD.** The overall design follows MonoDistill. First, we train the teacher network offline with the processed GT depth, which shares the same architecture as the student. Second, we load and freeze the teacher network and guide the student network with the teacher's features and responses. In the inference phase, we only use the parameters of the student network.

feature maps respectively,

$$r_{scc}(s,t) = \frac{\mathrm{Cov}(\mathrm{R}(s),(\mathrm{R}(t))}{\mathrm{Std}((\mathrm{R}(s))\mathrm{Std}((\mathrm{R}(t))} = \frac{\sum_{i=1}^{B}(\mathrm{R}(s_i)-\overline{\mathrm{R}}(s))(\mathrm{R}(t_i)-\overline{\mathrm{R}}(t))}{\sqrt{\sum_{i=1}^{B}(\mathrm{R}(s_i)-\overline{\mathrm{R}}(s))^2}\sqrt{\sum_{i=1}^{B}(\mathrm{R}(t_i)-\overline{\mathrm{R}}(t))^2}} \tag{2}$$

where B denotes batch size, $\mathrm{R}(s_i)$ is rank index of s_i, Cov(R(s)) is the covariance of $\mathrm{R}(s)$, $\overline{\mathrm{R}}(s)$ and $\mathrm{Std}((\mathrm{R}(s))$ denote the mean and standard derivation of $\mathrm{R}(s)$, respectively.

However, to evaluate the ranking relationship between different features, the Spearman correlation coefficient requires a ranking operation, which is not differentiable. Fortunately, Blondel et al. [2] introduced a novel method for fast differentiable sorting and ranking, so we adopted it.

3.3 Selection of Distillation Objects

Although MonoDistill proposes a practical monocular 3D detection knowledge distillation framework, it still faces the problem of excessive GPU memory usage. Specifically, in order to more effectively distill the output features of the backbone, MonoDistill introduces an attention-based feature fusion module. We argue that this strategy is inefficient and redundant. In contrast, we directly select the multi-scale feature maps of the neck as the distillation objects to deal with rich spatial information because the neck has the function of feature fusion. Another empirical explanation for choosing the feature map output from the neck as the object of distillation is that shallow features are more susceptible to noise than high-level features. Removing the heavy fusion module can save 30% GPU memory and speed up training. In particular, to introduce more information to aid in distillation, we keep the first layer output of the neck for distillation, which MonoDistill discards. In our experiments, selecting the multi-scale features of the neck for distillation can also improve performance (see Table 2 for more details). In Figure 4, we show the feature fusion process of the neck and the selected objects for distillation.

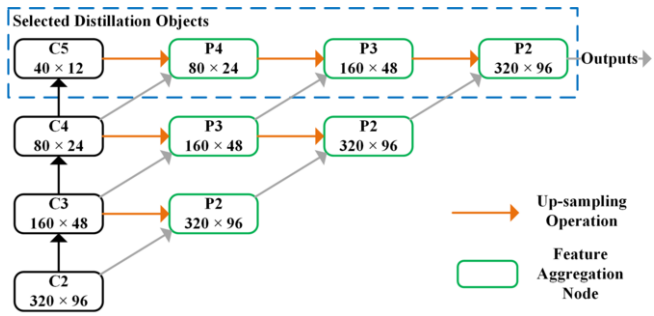

Figure 4: **Illustration of our distillation object selection strategy.** The green rectangle and orange arrow denote the feature aggregation node and up-sampling operation. The feature maps in the blue rectangle will be distilled.

3.4 Other Knowledge Distillation Strategy

By proposing a distillation strategy using the Spearman correlation coefficient to mine the ranking knowledge of networks across different modalities, we alleviate the misleading training problem faced by strict feature alignment distillation. However, the knowledge learned by the student model still needs to be improved to support the performance requirements of the 3D detector because the ranking relationship cannot fully represent the similarity between features. In order to obtain high-performance detectors, we still need a strict relational distillation to achieve high feature similarity in advance. We find that strict relation distillation combined with loose Spearman distillation can transfer the dark knowledge of the teacher model more effectively. Based on the above considerations, we retain the scene relation distillation in MonoDistill and let it play the role of strict relational distillation. Define f_i and f_j denote the i^{th} and j^{th} feature vector, and we calculate the similarity map as follows:

$$S_{i,j} = \frac{f_i^{\mathrm{T}} f_j}{\|f_i\|_2 \cdot \|f_j\|_2} \tag{3}$$

After that, we utilize L1 loss to train similarity maps of the teacher and student network.

$$\mathcal{L}_{sd} = \frac{1}{K \times K}\sum_{i=1}^{K}\sum_{j=1}^{K}\|S_{i,j}^{t} - S_{i,j}^{s}\|_1 \tag{4}$$

Table 1: **Quantitative comparisons of the Car category on the KITTI *testing* set.** The best results are listed in **red** and the second in **blue**. Note that DD3D employs the large private DDAD15M dataset (containing approximately 15M frames).

Methods	Venue	Extra Data	Runtime	AP_{3D} (Car test)			AP_{BEV} (Car test)		
				Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDLE [24]	CVPR 2021	None	40ms	17.23	12.26	10.29	24.79	18.89	16.00
MonoEF [44]	CVPR 2021	None	30ms	21.29	13.87	11.71	29.03	19.70	17.26
MonoFlex [42]	CVPR 2021	None	35ms	19.94	12.89	12.07	28.23	19.75	16.89
MonoRCNN [34]	ICCV 2021	None	70ms	18.36	12.65	10.03	25.48	18.11	14.10
GUPNet [22]	ICCV 2021	None	34ms	22.26	15.02	13.12	30.29	21.19	18.20
MonoCon [20]	AAAI 2022	None	26ms	22.50	16.46	13.95	31.12	22.10	19.00
Kinematic3D [3]	ECCV 2020	Temporal	120ms	19.07	12.72	9.17	26.69	17.52	13.10
AutoShape [21]	ICCV 2021	CAD	40ms	22.47	14.17	11.36	30.66	20.08	15.95
PatchNet [23]	ECCV 2020	LiDAR	400ms	15.68	11.12	10.17	22.97	16.86	14.97
D4LCN [10]	CVPR 2020	LiDAR	200ms	16.65	11.72	9.51	22.51	16.02	12.55
DDMP-3D [37]	CVPR 2021	LiDAR	180ms	19.71	12.78	9.80	28.08	17.89	13.44
CaDDN [31]	CVPR 2021	LiDAR	630ms	19.17	13.41	11.46	27.94	18.91	17.19
MonoDTR [18]	CVPR 2022	LiDAR	37ms	21.99	15.39	12.73	28.59	20.38	17.14
MonoDistill [9]	ICLR 2022	LiDAR	40ms	22.97	16.03	13.60	31.87	22.59	19.72
DID-M3D [29]	ECCV 2022	LiDAR	40ms	24.40	16.29	13.75	32.95	22.76	19.83
DD3D [27]	ICCV 2021	External	-	23.22	16.34	14.20	32.35	23.41	20.42
CMKD [16]	ECCV 2022	External	630ms	**25.09**	16.99	**15.30**	33.69	23.10	**20.67**
MonoSKD+MonoDLE	-	LiDAR	40ms	24.75	**17.07**	14.41	**34.43**	**23.62**	**20.59**
Improvements (to baseline)				+7.52	+4.81	+4.12	+9.64	+4.73	+4.59
MonoSKD+DID-M3D	-	LiDAR	40ms	**28.43**	**17.35**	**15.01**	**37.12**	**24.08**	20.37
Improvements (to baseline)				+4.03	+1.06	+1.26	+4.17	+1.32	+0.54

Where $\mathcal{L}_{sd}$, K, s, t represent the scene distillation loss, the number of feature vectors, and the student and teacher feature maps, respectively.

Furthermore, like most distillation works, we use the teacher's predictions as soft labels to guide the student network training. The object response distillation loss can be formulated as follows:

$$\mathcal{L}_{od} = \frac{1}{N}\sum_{k=1}^{N} \|M_g(p_k^s - p_k^t)\|_1 \tag{5}$$

where N, M_g, p_k^s and p_k^t represent the number of detection heads, the Gaussian-like mask, and prediction of the k^{th} detection head from student and teacher network.

3.5 End-to-end Training

For the convenience of expression, we define the inherited losses from the student network as $\mathcal{L}_{reg}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{dep}$ for bounding box regression, object classification, and depth regression, respectively. The total training loss is:

$$\mathcal{L} = \mathcal{L}_{reg} + \mathcal{L}_{cls} + \mathcal{L}_{dep} + \mathcal{L}_{od} + \mathcal{L}_{sd} + \alpha\mathcal{L}_{scc} \tag{6}$$

where α are hyper-parameters to balance the detection training loss and distillation loss. Rather than specially selecting the optimal hyper-parameters, we choose $\alpha = 1$ for simplicity.

4 Experiments

4.1 Dataset and Metrics

Following the previous works [22, 24], we perform experiments on the challenging KITTI [11] dataset. The KITTI dataset comprises 7,481 training samples and 7,518 testing samples, where the labels of training samples are publicly available, and the labels of testing samples are private, which are only used for online evaluation and ranking. To conduct ablations, we further divide the training samples into a train set (3,712 samples) and a validation set (3,769 samples), following prior works [7]. The final results of our method are reported on *testing* set, while the ablation studies are conducted on the *validation* set. Besides, KITTI divides objects into *easy*, *moderate*, and *hard* levels according to the 2D box height, occlusion, and truncation levels of one object. In KITTI, only three categories of performance are mainly concerned: car, pedestrian, and cyclist, among which the performance of **car** with the **moderate** level is the most critical. For evaluation metrics, both 3D detection and Bird's Eye View (BEV) detection are evaluated using the AP_{40} metric.

4.2 Implementation Details

To demonstrate the generalizability and effectiveness of our method, we choose three monocular 3D detection networks for distillation experiments: MonoDLE, GUPNet, and DID-M3D. We accomplish our method with the PyTorch framework [28]. Taking the MonoDLE detector as an example, we conduct experiments on 2 NVIDIA RTX 3090 GPUs with batch size 12 and train it for 160 epochs, which takes almost 10 hours. See Appendix A.2 for experimental details of GUPNet and DID-M3D. We choose the Adam optimizer with the initial learning rate $1e^{-5}$. We apply the linear warm-up strategy for the first five epochs of training, which increases the learning rate to $1e^{-3}$. Afterward, the learning rate decays in epochs 95 and 125 with a rate of 0.1. For the backbone, neck, and head, we follow the design of MonoDistill. We train the teachers with the same dense depth maps used in MonoDistill for a fair comparison. To execute the distillation process, we have pre-trained the teacher network, and the teacher performance is demonstrated in Appendix Table 9. Additionally, our code will be open-sourced for reproducibility.

4.3 State-of-the-art Comparisions

As is shown in Table 1, we compare the experimental results of our framework and other state-of-the-art methods on the KITTI *testing* set. Our schemes are significantly improved compared to

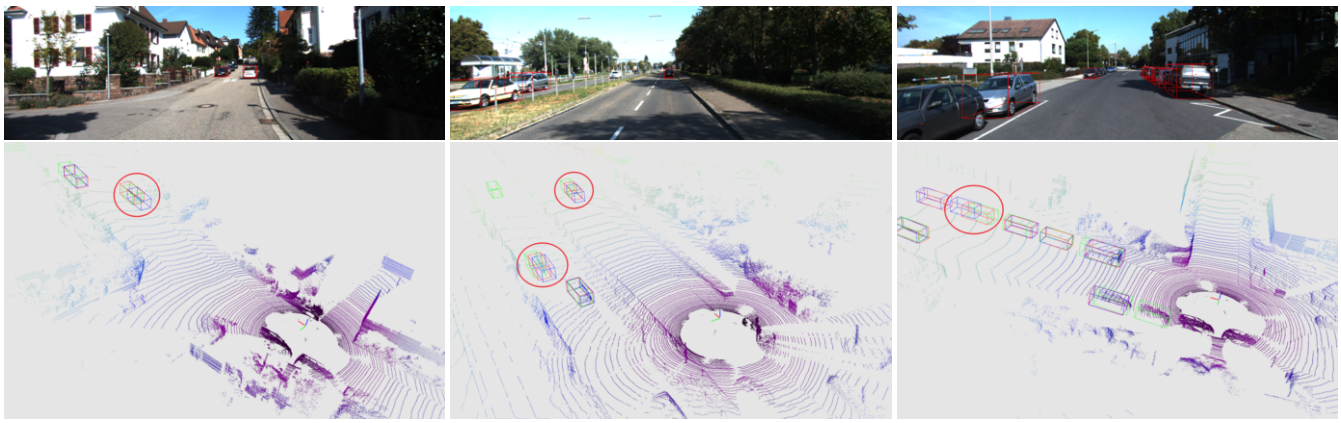

Figure 5: **Qualitative results.** We employ blue, red, and green 3D boxes to denote the DID-M3D baseline, MonoSKD, and ground truth results. Additionally, we use red circles to highlight significant differences.

Table 2: **Ablation studies on the KITTI validation set.** We conduct experiments based on the **MonoDLE** network. SD, SCC, and OD denote the scene distillation, the Spearman distillation, and the object response distillation, respectively. †: Distilled object selection strategy proposed in Section 3.3.

#	SD	SCC	OD	AP_{3D} (Car val) Easy	Mod.	Hard	AP_{BEV} (Car val) Easy	Mod.	Hard
MonoDistill				24.40	18.47	16.46	32.86	25.14	21.99
Our MonoDistill†				24.67	18.64	15.74	34.40	25.81	22.45
(a)				19.86	15.11	12.64	26.93	21.03	18.33
(b)	✓			21.35	16.97	14.59	29.54	22.71	19.69
(c)		✓		21.26	16.93	14.47	29.16	22.44	19.53
(d)			✓	22.23	17.60	15.02	31.47	23.75	21.46
(e)	✓	✓		21.41	17.17	14.67	29.71	22.97	20.04
(f)	✓		✓	24.74	18.44	15.63	33.83	25.18	21.90
(g)		✓	✓	24.63	18.32	15.49	33.30	25.14	21.82
(h)	✓	✓	✓	**26.10**	**19.18**	**16.96**	**34.77**	**25.75**	**22.44**
				+6.24	+4.07	+4.32	+7.84	+4.72	+4.11

the respective baseline models and outperform other state-of-the-art methods. On the car category that KITTI cares most about, our MonoSKD+DID-M3D results achieve state-of-the-art performance on almost all 3D-level and BEV-level metrics. In particular, compared with the second-best results, our scheme achieves up to 13.31% and 7.81% relative performance improvements on 3D-level and BEV-level metrics. Our scheme can theoretically be applied to any monocular 3D detector and improve performance.

4.4 *Ablation Study*

In this section, we analyze the effectiveness of each part of our distillation framework on the KITTI *validation* set. As shown in Table 2, considering that GUPNet and DID-M3D uses the ROI-align operation and the reproducibility cannot be guaranteed, we conduct the ablation experiment base on the MonoDLE network. It is noteworthy that we conduct our ablation experiments on the redesigned distillation framework (discussed in Section 3.3). Our redesigned distillation framework is higher than the original MonoDistill in most indicators, which has lower GPU memory occupation and faster training speed (see Appendix A.3). Specifically, all distillation schemes improve the accuracy of the baseline model, and their improvements are complementary. Compared with the baseline, the final model can improve the **absolute** detection performance by **6.24**, **4.07**, **4.32** and **absolute** BEV performance by **7.84**, **4.72**, **4.11** on the easy, moderate, and hard levels, respectively.

4.5 *Distillation with Different Detectors and Backbones*

To further demonstrate the effectiveness and generalizability of our method, we select three monocular 3D detection networks and compare our method with the competitive MonoDistill scheme. As shown in Table 3, our distillation scheme has achieved better performance than MonoDistill on all detectors in terms of detection performance and BEV performance, which also verifies the superiority of our scheme.

In Table 4, we select representative backbones like ResNet [14] and MobileNetv3 [17] to supplement related experiments, and the results show that our scheme is not sensitive to the backbones.

Table 3: **Distillation performance with different detectors of the Car category on the KITTI *validation* set.** †: our reproduced results, whose performance is much higher than that reported in the original MonoDistill paper.

Methods	AP_{3D} (Car val) Easy	Mod.	Hard	AP_{BEV} (Car val) Easy	Mod.	Hard
MonoDLE	19.86	15.11	12.64	26.93	21.03	18.33
+MonoDistill	24.40	18.47	16.46	32.86	25.14	21.99
+Ours	**26.10**	**19.18**	**16.96**	**34.77**	**25.75**	**22.44**
GUPNet	21.19	16.23	13.57	30.14	22.38	19.29
+MonoDistill†	24.34	17.72	14.89	31.74	23.22	19.98
+Ours	**25.30**	**18.06**	**15.37**	**32.54**	**23.72**	**21.71**
DID-M3D	25.75	17.77	14.74	33.39	23.66	20.86
+MonoDistill	27.08	19.31	16.16	35.85	25.47	21.73
+Ours	**28.91**	**20.21**	**16.99**	**37.66**	**26.41**	**23.39**

4.6 *Qualitative Results*

In order to demonstrate the superiority of our method more intuitively, we visualize the results predicted by the network. As shown in Figure 5, we apply the red boxes to represent the result of our proposed MonoSKD+DID-M3D. Our model has better localization performance than the baseline model.

4.7 *Pedestrian/Cyclist Detection*

To demonstrate the generalizability of other categories, we perform experiments on cyclist and pedestrian categories (see Table 5). We

Table 4: **Quantitative comparison of different backbones on KITTI *validation* set.** We conduct experiments based on MonoDLE network. The best results are listed in **bold**.

Backbones	AP_{3D} (Car val)			AP_{BEV} (Car val)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Res18 (T)	66.54	48.31	41.56	77.78	61.60	52.96
Res18 (S)	17.96	13.91	12.17	24.78	18.74	16.72
+MonoDistill	20.25	15.60	13.12	27.88	20.83	17.95
+MonoSKD	**21.58**	**16.31**	**13.70**	**28.52**	**21.55**	**18.48**
MobileNetv3-L (T)	65.77	48.55	40.81	77.36	60.75	52.19
MobileNetv3-L (S)	15.73	12.37	10.33	23.47	17.57	15.67
+MonoDistill	16.59	12.93	10.71	25.34	19.58	16.81
+MonoSKD	**18.15**	**14.02**	**12.28**	**26.97**	**20.24**	**17.34**

use the pre-trained model provided by DID-M3D as the baseline and retrain the three-category teacher model. Our scheme achieves the best results in almost all indicators. Moreover, we provide the full three-category performance in Appendix Table 10. It is worth noting that the performance of the **Cyclist** in MonoDistill actually drops after distillation. We guess this is because the bounding boxes of the Cyclist category often contain more background pixels, so the alignment based on L1 loss misleads learning. In contrast, our Spearman distillation scheme is more relaxed and steadily improves detection performance.

Table 5: **Performance of Pedestrian/Cyclist detection on the KITTI *validation* set under IoU criterion 0.5.** The best results are listed in bold.

Methods	AP_{3D}(Pedestrian)			AP_{3D}(Cyclist)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
DID-M3D	11.15	8.65	7.15	5.40	2.81	2.72
+MonoDistill	15.96	11.86	**9.80**	4.73	2.69	2.50
+MonoSKD	**16.24**	**11.92**	9.28	**5.50**	**3.45**	**3.00**

4.8 *Sensitivity study of loss weight α*

In Eq. 6, we use the loss weight hyper-parameter α to balance the detection training loss and distillation loss. Here, we conduct several experiments to investigate the influence of α. As shown in Table 6, the worst result is just a 0.38 mAP drop compared with the best result (19.18 $\rightarrow$ 18.80), indicating our method is not sensitive to the hyper-parameter α.

Table 6: **Ablation study of loss weight hyper-parameter α.**

α	AP_{3D} (Car val)			AP_{BEV} (Car val)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
0.5	25.10	18.80	16.58	34.42	25.60	22.16
1.0	26.10	**19.18**	**16.96**	34.77	**25.75**	**22.44**
2.0	**26.38**	19.05	16.81	**34.81**	25.68	22.38

4.9 *Insights into Spearman Distillation and Downstream Task Generality*

Spearman distillation introduces a loose constraint between cross-modal features instead of strict alignment and is more suitable for cross-modal tasks. The 3D object detection task is a typical cross-model task for its multiple modalities (e.g., Camera, LiDAR, and Radar) inputs, and thus, we choose monocular 3D object detection task to verify our method. We can put the whole story on the knowledge distillation itself. Spearman distillation has strong knowledge transfer potential and can be easily extended to downstream tasks. Take 2D detection as an example (Table 7). SKD can achieve significant performance gains without scene relation distillation, even with **heterogeneous** teachers. CMKD is a distillation method for BEV paradigm 3D detection, so we replaced the MSE loss in CMKD with the proposed SKD. Table 8 reveals that SKD suits the BEV paradigm detectors. Compared with CMKD, SKD brings performance improvements in almost all indicators.

Table 7: **Distilling Student Detectors with Homogeneous and Heterogeneous Teachers on the COCO dataset.**

Method	schedule	mAP	AP_S	AP_M	AP_L
Retina-ResX101 (T)	2x	40.8	22.9	44.5	54.6
Retina-Res50 (S)	2x	37.4	20.0	40.7	49.7
+FRS [43]	2x	40.1	21.9	43.7	54.3
+FGD [39]	2x	40.4	**23.4**	44.7	54.1
+SKD (Ours)	2x	**40.6**	22.0	**44.8**	**54.8**
FCOS-X101 (T)	2x+ms	42.7	26.0	46.5	54.7
Retina-Res50 (S)	1x	36.5	20.4	40.3	48.1
+SKD (Ours)	1x	**40.1**	**22.7**	**44.3**	**53.8**

Table 8: **Effectiveness experiments under the BEV paradigm.**

Method	AP_{3D} (Car val)			AP_{BEV} (Car val)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
CMKD [16]	23.49	15.56	12.97	**31.81**	21.02	18.57
+SKD (Ours)	**23.59**	**15.79**	**13.16**	31.79	**22.27**	**19.05**

5 Conclusion

In this paper, we propose **MonoSKD**, a cross-modal distillation framework for monocular 3D object detection via Spearman's rank correlation coefficient. Existing distillation schemes try to strictly align cross-modal features, thus leading to suboptimal distillation performance. To alleviate this problem, we propose to use the Spearman correlation coefficient to help mine ranking knowledge among features. To improve distillation efficiency, we select the appropriate distillation objects to save 30% of GPU memory and accelerate training. We distill three detectors to verify the effectiveness of our scheme and achieve state-of-the-art performance on the challenging KITTI benchmark without introducing additional inference costs.

References

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen, 'Pearson correlation coefficient', in *Noise reduction in speech processing*, 1–4, Springer, (2009).

[2] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga, 'Fast differentiable sorting and ranking', in *International Conference on Machine Learning*, pp. 950–959. PMLR, (2020).

[3] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele, 'Kinematic 3d object detection in monocular video', in *European Conference on Computer Vision*, pp. 135–152. Springer, (2020).

[4] Weihan Cao, Yifan Zhang, Jianfei Gao, Anda Cheng, Ke Cheng, and Jian Cheng, 'Pkd: General distillation framework for object detectors via pearson correlation coefficient', *arXiv preprint arXiv:2207.02039*, (2022).

[5] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau, 'Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2040–2049, (2017).

[6] 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).

[7] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun, '3d object proposals using stereo imagery for accurate object class detection', *IEEE transactions on pattern analysis and machine intelligence*, **40**(5), 1259–1272, (2017).

[8] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li, 'Monopair: Monocular 3d object detection using pairwise spatial relationships', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (June 2020).

[9] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang, 'Monodistill: Learning spatial features for monocular 3d object detection', *arXiv preprint arXiv:2201.10830*, (2022).

[10] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo, 'Learning depth-guided convolutions for monocular 3d object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1000–1001, (2020).

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun, 'Are we ready for autonomous driving? the kitti vision benchmark suite', in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, (2012).

[12] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu, 'Distilling object detectors via decoupled features', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2154–2164, (2021).

[13] 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*, pp. 3153–3163, (2021).

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, (2016).

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

[16] Yu Hong, Hang Dai, and Yong Ding, 'Cross-modality knowledge distillation network for monocular 3d object detection', in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pp. 87–104. Springer, (2022).

[17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al., 'Searching for mobilenetv3', in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, (2019).

[18] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu, 'Monodtr: Monocular 3d object detection with depth-aware transformer', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4012–4021, (2022).

[19] 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*, pp. 12697–12705, (2019).

[20] Xianpeng Liu, Nan Xue, and Tianfu Wu, 'Learning auxiliary monocular contexts helps monocular 3d object detection', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1810–1818, (2022).

[21] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang, 'Autoshape: Real-time shape-aware monocular 3d object detection', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15641–15650, (2021).

[22] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang, 'Geometry uncertainty projection network for monocular 3d object detection', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3111–3121, (2021).

[23] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang, 'Rethinking pseudo-lidar representation', in *European Conference on Computer Vision*, pp. 311–327. Springer, (2020).

[24] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang, 'Delving into localization errors for monocular 3d object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4721–4730, (2021).

[25] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon, 'Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2069–2078, (2019).

[26] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka, '3d bounding box estimation using deep learning and geometry', in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7074–7082, (2017).

[27] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon, 'Is pseudo-lidar needed for monocular 3d object detection?', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3142–3152, (2021).

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., 'Pytorch: An imperative style, high-performance deep learning library', *Advances in neural information processing systems*, **32**, (2019).

[29] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai, 'Did-m3d: Decoupling instance depth for monocular 3d object detection', in *European Conference on Computer Vision*, (2022).

[30] Zengyi Qin, Jinglu Wang, and Yan Lu, 'Monogrnet: A general framework for monocular 3d object detection', *IEEE transactions on pattern analysis and machine intelligence*, (2021).

[31] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander, 'Categorical depth distribution network for monocular 3d object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8555–8564, (2021).

[32] Thomas Roddick, Alex Kendall, and Roberto Cipolla, 'Orthographic feature transform for monocular 3d object detection', *arXiv preprint arXiv:1811.08188*, (2018).

[33] 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).

[34] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim, 'Geometry-based distance decomposition for monocular 3d object detection', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15172–15181, (2021).

[35] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen, 'Channel-wise knowledge distillation for dense prediction', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5311–5320, (2021).

[36] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel, 'Mvx-net: Multimodal voxelnet for 3d object detection', in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7276–7282. IEEE, (2019).

[37] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang, 'Depth-conditioned dynamic message propagation for monocular 3d object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 454–463, (2021).

[38] 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*, pp. 4933–4942, (2019).

[39] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan, 'Focal and global knowledge distillation for detectors', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4643–4652, (2022).

[40] Linfeng Zhang, Runpei Dong, Hung-Shuo Tai, and Kaisheng Ma, 'Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection', *arXiv preprint arXiv:2205.11098*, (2022).

[41] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li, 'Monodetr: Depth-aware transformer for monocular 3d object detection', *arXiv preprint arXiv:2203.13310*, (2022).

[42] Yunpeng Zhang, Jiwen Lu, and Jie Zhou, 'Objects are different: Flexible monocular 3d object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3289–3298, (2021).

[43] Du Zhixing, Rui Zhang, Ming Chang, Shaoli Liu, Tianshi Chen, Yunji Chen, et al., 'Distilling object detectors with feature richness', *Advances in Neural Information Processing Systems*, **34**, 5213–5224, (2021).

[44] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang, 'Monocular 3d object detection: An extrinsic parameter free approach', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7556–7566, (2021).

A Appendix

Considering the space constraints of the main text, we provide more experimental results and discussions in the supplementary material.

A.1 Motivation

In recent years, remarkable progress has been made in monocular 3D object detection. However, these lightweight detectors face the problem of low detection performance, so distillation frameworks such as MonoDistill are proposed to alleviate this problem. In our research on MonoDistill, we find that the distillation technique used by MonoDistill is a stricter constraint based on the 2D detection distillation scheme. Because 2D object detection uses RGB image input, the difference between the teacher and student models is not vast, so that we can use a strict distillation strategy. When it comes to monocular 3D object detection, there is a vast difference in the input of the teacher and the student model. Directly aligning features may mislead the training, so finding a general and loose distillation strategy is necessary. Specifically, we try to distill the relative relationship between features, introducing the Spearman correlation coefficient.

Table 9: **Quantitative comparison between teacher and student on the KITTI *validation* set.** 'T' indicates the teacher using the dense depth maps as input for training and inference. 'S' indicates the student without distillation.

Methods	AP_{3D} (Car val)			AP_{BEV} (Car val)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDLE (T)	60.57	45.06	37.90	74.20	57.49	50.89
MonoDLE (S)	19.86	15.11	12.64	26.93	21.03	18.33
GUPNet (T)	48.54	32.88	27.44	61.29	43.58	37.63
GUPNet (S)	21.19	16.23	13.57	30.14	22.38	19.29
DID-M3D (T)	63.71	43.81	36.97	74.59	55.09	46.52
DID-M3D (S)	25.75	17.77	14.74	33.39	23.66	20.86

A.2 More Details of Our Experiments

To execute the distillation process, we have pre-trained the teacher network, and the performance of teacher and student is demonstrated in Table 9.

MonoDLE. First, we chose MonoDLE as the baseline model for our ablation studies because we noticed that GUPNet and DID-M3D apply ROI-align operation, which is irreproducible during training. We believe that this property affects the fairness of the experiment. In contrast, the MonoDLE network has good reproducibility, so we do ablation experiments based on MonoDLE. The same settings as MonoDistill are used when choosing MonoDLE as the baseline: learning rate, optimizer, and batch size.

GUPNet. Because MonoDistill does not open-source the distillation code of GUPNet, we reproduce the relevant results, and our distillation results have higher performance. In addition, the authors of GUPNet open-source the pre-trained model, so we directly reuse it as a student model. Because we use a different PyTorch version, it is reasonable that the performance of the pre-trained models is slightly different. For GUPNet, we use three categories in KITTI for training.

DID-M3D. Following the original author's training setting, the DID-M3D results we report in the main text are all trained in the 'Car' category. Again, we keep the same training settings as the original paper. It is worth noting that when training the teacher models of MonoDLE and GUPNet, the GT depth map we use is provided by MonoDistill. However, DID-M3D has already provided a preprocessed depth map. At this time, we directly reuse the depth map provided by DID-M3D.

A.3 Advantages of the Redesigned Framework

To validate the effectiveness of our distillation framework, we report the average GPU memory and training time per epoch for MonoDLE and DID-M3D, respectively. As is shown in Figure 6, compared with MonoDistill, our distillation framework saves at least 30% of GPU memory while bringing faster training. In addition, we quantitatively show that the new framework still brings performance improvements in ablation experiments (see Table 2).

A.4 Detailed Car/Pedestrian/Cyclist Detection

Compared to the 'Car' category, the 'Pedestrian' and 'Cyclist' categories have small sizes, non-rigid structures, and limited training samples, so they are much more challenging to detect. As shown in Table 10, we report the full detection and BEV performance for Pedestrian, Cyclist and Car categories, and we can see that our scheme is still effective. To avoid ambiguity, we clarify that DID-M3D is only trained in the Car category in the main text, so Car's performance will be slightly higher. In order to compare with the pre-trained model provided by DID-M3D, we retrain the models of three categories.

Observing Table 10, our MonoSKD is better on almost all metrics but slightly weaker on 'Hard' metrics for the pedestrian category. It is reasonable because we scale the feature map during distillation to speed up the convergence of Spearman loss, so the pedestrian category does not get enough training due to its smaller size. In addition, we find that our scheme has apparent advantages at almost all 'Easy' and 'Mod.' levels, proving our point of view from the side. We will talk about this in the next section.

A.5 Weaknesses of Our Method

Although our scheme has achieved performance improvements on the KITTI *validation* set and *testing* set, it cannot be ignored that our scheme still has some shortcomings.

We adapt the scaling strategy because the detection task is a dense prediction task, so too many feature pixels lead to a decrease in sorting efficiency. In order to speed up the training, we adapt the scaling strategy. Therefore, small objects that originally accounted for a low proportion of the feature map may disappear entirely after scaling. Consequently, our method does not all outperform baselines, especially in 'Hard' settings.

Suggestions. Limited by the currently almost unsolvable sorting efficiency problem, we recommend that when using Spearman distillation, determine the feature map size in the distillation process according to the data set or only distill the object area in the feature map. In our experiments, we find that by increasing the size of the feature map during Spearman distillation or focusing on the features of the object region without scaling, performance improvement can be obtained. For the former, if the categories are large objects (such as "car" in KITTI), relatively small feature map sizes can be used to speed up training. Otherwise, training time will be sacrificed. For the latter, we only need to use the features of the region of interest without scaling. Considering efficiency issues, our experimental results in the main text do not adopt these suggestions.

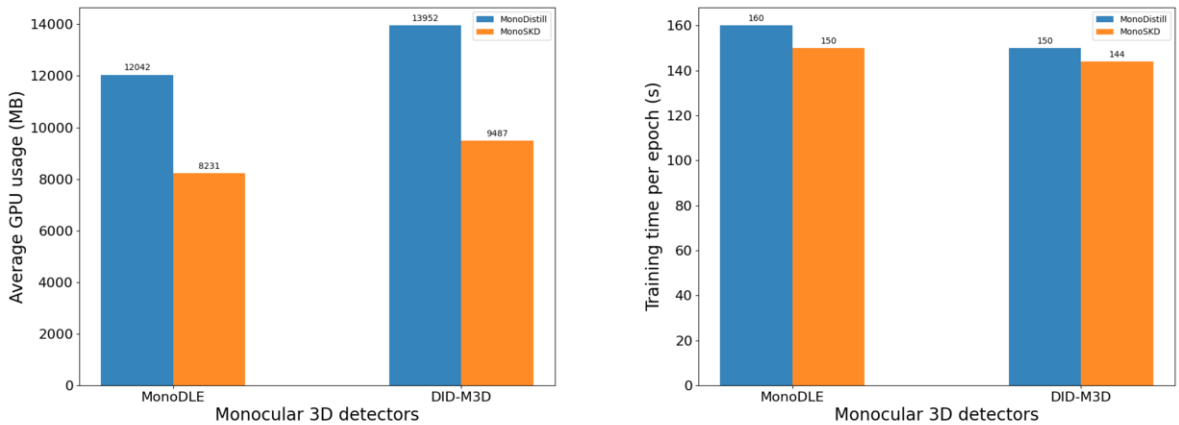

Figure 6: **Average GPU usage and training time of the redesigned distillation framework.** We use two GPUs, and the batch size of MonoDLE is 12, and the batch size of DID-M3D is 16.

Table 10: **Performance of Car/Pedestrian/Cyclist detection on the KITTI *validation* set.** Please note, Pedestrian/Cyclist performance is calculated under IoU criterion 0.5.

Methods	AP_{3D} (Car)			AP_{BEV} (Car)			AP_{3D} (Ped.)			AP_{BEV} (Ped.)			AP_{3D} (Cyclist)			AP_{BEV} (Cyclist)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
DID-M3D	25.36	17.03	14.05	33.90	23.25	19.51	11.15	8.65	7.15	12.84	10.23	7.90	5.40	2.81	2.72	5.97	3.50	3.02
+MonoDistill	26.14	18.08	14.84	34.50	24.56	20.84	15.96	11.86	**9.80**	17.84	**13.64**	**11.42**	4.73	2.69	2.50	5.31	2.82	2.74
+MonoSKD	**27.53**	**18.25**	**14.96**	**36.15**	**25.08**	**21.14**	**16.24**	**11.92**	9.28	**18.43**	13.60	10.77	**5.50**	**3.45**	**3.00**	**6.48**	**3.76**	**3.59**

Table 11: **Comparing Pearson and Spearman correlation coefficients on the KITTI *validation* set.** We choose **MonoDLE** as baseline model. 'PCC' stands for Pearson Correlation Coefficient, and 'SCC' stands for Spearman Correlation Coefficient. The best results are listed in bold.

Methods	AP_{3D}(Car val)			AP_{BEV}(Car val)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDistill	24.40	18.47	16.46	32.86	25.14	21.99
with PCC	25.08	18.69	16.58	33.66	25.30	22.11
with SCC	**26.10**	**19.18**	**16.96**	**34.77**	**25.75**	**22.44**

A.6 Discussion about Pearson Knowledge Distillation

Our scheme uses the Spearman correlation coefficient. Someone may wonder why we do not select the Pearson correlation coefficient for distillation. This section focuses on applying the Pearson correlation coefficient in cross-modal distillation.

First of all, PKD-based distillation strategies are not suitable for cross-modal tasks. Although the normalization operation can further alleviate the feature difference, this method is not the optimal solution under the premise of a vast modal difference. In the 2D object detection task, due to the input data's consistent modality, the whole task's distillation is relatively easy, even for heterogeneous 2D detectors.

Besides, PKD is a particular distillation method that considers both relation-based and feature-based distillation. We think PKD is more inclined to feature-based distillation and replace the feature map L1 loss in MonoDistill with PKD. As is shown in Table 11, the PCC strategy can achieve a certain performance improvement in the ***validation*** set, but the improvement is very limited. In contrast, our scheme brings a more obvious performance improvement. In Table 12, we show the performance results of our scheme and PKD on the KITTI ***testing*** set. The results show that our method significantly outperforms PKD.

Table 12: **Comparing Pearson and Spearman correlation coefficients on the KITTI *testing* set.** We choose **MonoDLE** and **DID-M3D** as baseline model. The best results are listed in bold.

Methods	AP_{3D}(Car test)			AP_{BEV}(Car test)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
PKD+MonoDLE	23.39	16.51	14.08	31.92	22.03	19.90
MonoSKD+MonoDLE	**24.75**	**17.07**	**14.41**	**34.43**	**23.62**	**20.59**
PKD+DID-M3D	26.55	16.89	14.74	36.01	23.71	20.25
MonoSKD+DID-M3D	**28.43**	**17.35**	**15.01**	**37.12**	**24.08**	**20.37**