X^3KD: Knowledge Distillation Across Modalities, Tasks and Stages for Multi-Camera 3D Object Detection

Marvin Klingner [*,†] Shubhankar Borse [*,‡] Varun Ravi Kumar [*,†]
Behnaz Rezaei [†] Venkatraman Narayanan [†] Senthil Yogamani [§] Fatih Porikli [‡]
{mklingne, sborse, vravikum, brezaei, vennara, syogaman, fporikli}@qti.qualcomm.com

Abstract

Recent advances in 3D object detection (3DOD) have obtained remarkably strong results for LiDAR-based models. In contrast, surround-view 3DOD models based on multiple camera images underperform due to the necessary view transformation of features from perspective view (PV) to a 3D world representation which is ambiguous due to missing depth information. This paper introduces X^3KD, a comprehensive knowledge distillation framework across different modalities, tasks, and stages for multi-camera 3DOD. Specifically, we propose cross-task distillation from an instance segmentation teacher (X-IS) in the PV feature extraction stage providing supervision without ambiguous error backpropagation through the view transformation. After the transformation, we apply cross-modal feature distillation (X-FD) and adversarial training (X-AT) to improve the 3D world representation of multi-camera features through the information contained in a LiDAR-based 3DOD teacher. Finally, we also employ this teacher for cross-modal output distillation (X-OD), providing dense supervision at the prediction stage. We perform extensive ablations of knowledge distillation at different stages of multi-camera 3DOD. Our final X^3KD model outperforms previous state-of-the-art approaches on the nuScenes and Waymo datasets and generalizes to RADAR-based 3DOD. Qualitative results video at https://youtu.be/1do9DPFmr38.

1. Introduction

3D object detection (3DOD) is an essential task in various real-world computer vision applications, especially autonomous driving. Current 3DOD approaches can be categorized by their utilized input modalities, *e.g.*, camera images [28, 40, 46] or LiDAR point clouds [25, 55, 60], which dictates the necessary sensor suite during inference. Recently, there has been significant interest in surround-view

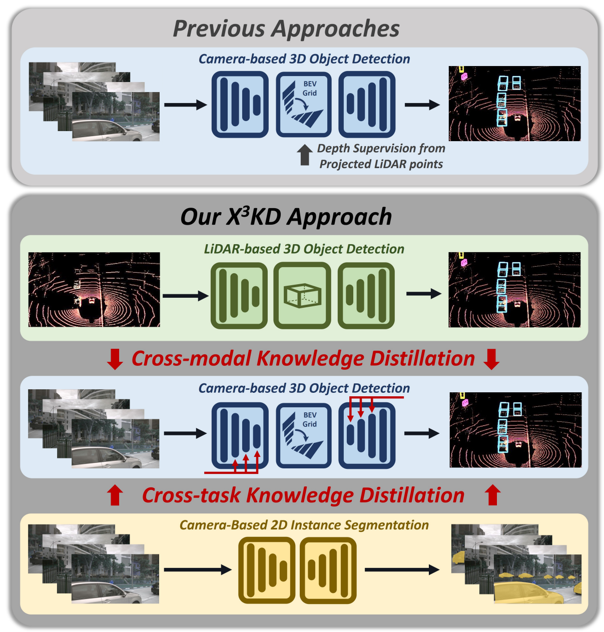

Figure 1. While previous approaches considered multi-camera 3DOD in a standalone fashion or with depth supervision, **we propose X^3KD, a knowledge distillation framework using cross-modal and cross-task information** by distilling information from LiDAR-based 3DOD and instance segmentation teachers into different stages (marked by red arrows) of the multi-camera 3DOD.

multi-camera 3DOD, aiming to leverage multiple low-cost monocular cameras, which are conveniently embedded in current vehicle designs in contrast to expensive LiDAR scanners. Existing solutions to 3DOD are mainly based on extracting a unified representation from multiple cameras [28,30,37,41] such as the bird's-eye view (BEV) grid. However, predicting 3D bounding boxes from 2D perspective-view (PV) images involves an ambiguous 2D to 3D transformation without depth information, which leads to lower performance compared to LiDAR-based 3DOD [1, 28, 30, 55].

While LiDAR scanners may not be available in commercially deployed vehicle fleets, they are typically available in training data collection vehicles to facilitate 3D annotation. Therefore, LiDAR data is privileged; it is often available

*These authors contributed equally to this work.
†Automated Driving, Qualcomm Technologies, Inc.
‡Qualcomm AI Research, an initiative of Qualcomm Technologies, Inc.
§Automated Driving, QT Technologies Ireland Limited

Model	*LSS++*	*DS*	GFLOPS	*mAP*↑	*NDS*↑
BEVDepth†	✗	✗	298	32.4	44.9
	✗	✓	298	33.1	44.9
	✓	✗	316	34.9	47.0
	✓	✓	316	35.9	47.2
X^{3}KD (Ours)	✓	✓	316	**39.0**	**50.5**

Table 1. **Analysis of BEVDepth†** (re-implementation of [28]): We compare the architectural improvement of a larger Lift-Splat-Shoot (LSS++) transform to using depth supervision (DS).

during training but not during inference. The recently introduced BEVDepth [28] approach pioneers using accurate 3D information from LiDAR data at training time to improve multi-camera 3DOD, see Fig. 1 (top part). Specifically, it proposed an improved Lift-Splat-Shoot PV-to-BEV transform (LSS++) and depth supervision (DS) by projected LiDAR points, which we analyze in Table 1. We observe that the LSS++ architecture yields significant improvements, though depth supervision seems to have less effect. This motivates us to find additional types of supervision to transfer accurate 3D information from LiDAR point clouds to multi-camera 3DOD. To this end, we propose cross-modal knowledge distillation (KD) to not only use LiDAR *data* but a high-performing LiDAR-based 3DOD *model*, as in Fig. 1 (middle part). To provide an overview of the effectiveness of cross-modal KD at various multi-camera 3DOD network stages, we present three distillation techniques: feature distillation (X-FD) and adversarial training (X-AT) to improve the feature representation by the intermediate information contained in the LiDAR 3DOD model as well as output distillation (X-OD) to enhance output-stage supervision.

For optimal camera-based 3DOD, extracting useful PV features before the view transformation to BEV is equally essential. However, gradient-based optimization through an ambiguous view transformation can induce non-optimal supervision signals. Recent work proposes pre-training the PV feature extractor on instance segmentation to improve the extracted features [49]. Nevertheless, neural networks are subject to catastrophic forgetting [23] such that knowledge from pre-training will continuously degrade if not retained by supervision. Therefore, we propose cross-task instance segmentation distillation (X-IS) from a pre-trained instance segmentation teacher into a multi-camera 3DOD model, see Fig. 1 (bottom part). As shown in Table 1, our X^{3}KD framework significantly improves upon BEVDepth without additional complexity during inference.

To summarize, our main contributions are as follows:

- We propose X^{3}KD, a KD framework across modalities, tasks, and stages for multi-camera 3DOD.
- Specifically, we introduce cross-modal KD from a strong LiDAR-based 3DOD teacher to the multi-camera 3DOD student, which is applied at multiple network stages in bird's eye view, *i.e*., feature-stage (X-FD and X-AT) and output-stage (X-OD).
- Further, we present cross-task instance segmentation distillation (X-IS) at the PV feature extraction stage.
- X^{3}KD outperforms previous approaches for multi-camera 3DOD on the nuScenes and Waymo datasets.
- We transfer X^{3}KD to RADAR-based 3DOD and train X^{3}KD only through KD without using ground truth.
- Our extensive ablation studies on nuScenes and Waymo provide a comprehensive evaluation of KD at different network stages for multi-camera 3DOD.

2. Related Work

Multi-View Camera-Based 3D Object Detection: Current multi-view 3D object detectors can be divided into two main streams: First, DETR3D and succeeding works [30, 32, 33, 46, 59] project a sparse set of learnable 3D queries/priors onto 2D image features with subsequent sampling and an end-to-end 3D bounding box regression. Second, LSS and following works [2, 18, 28, 40] employ a view transformation consisting of a depth prediction, a point cloud reconstruction, and a voxel pooling to project points to BEV. 3D bounding boxes are predicted from these BEV features. While such works focus on improving the network architecture and view transformation, we focus on better model optimization. In this direction, M^{2}BEV [49] proposed segmentation [3, 4] pre-training of the PV feature extraction. We propose cross-task instance segmentation distillation to retain this knowledge during 3DOD training.

Most current state-of-the-art works focus on incorporating temporal information either through different kinds of feature-level aggregation [17, 28, 30, 33] or by improving depth estimation by temporal stereo approaches [27, 47]. While the usual setting considers data from 2 time steps, recently proposed SOLOFusion [38] separately models long-range and short-range temporal dependencies in input data from 16 time steps. Our work focuses on a different direction, *i.e*., we try to optimally exploit the information contained in LiDAR point clouds. In this direction, BEVDepth [28] and succeeding works [27,38] supervise the depth estimation with projected LiDAR points. We explore this path further by using cross-modal knowledge distillation (KD) from a LiDAR-based 3DOD teacher.

Multi-Modal 3D Object Detection: Recently, there has been a trend to fuse different sensor modalities, especially camera and LiDAR, with the idea of combining modality-specific useful information, hence improving the final 3DOD performance [1, 22, 29, 35, 50, 53]. Existing 3DOD methods mostly perform multi-modal fusion at one of the three stages: First, various approaches [44, 45, 50] propose to decorate/augment the raw LiDAR points with image features. Second, intermediate feature fusion of the modalities in a shared representation space, such as the BEV space, has been explored [8, 22, 29, 35, 53]. Third, proposal-based fusion methods [1,7,24] keep the feature extraction of different modalities independent and aggregate multi-modal features via proposals or queries in the 3DOD prediction head. While these approaches require both sensors to be available during inference, our X^{3}KD approach requires only camera sensors during inference. We also apply our KD approach to less frequently explored RADAR- and camera-RADAR fusion-based models.

Knowledge Distillation for 3D Object Detection: Employing the KD technique from [15] some recent works have explored KD for 3DOD [9, 31, 48, 56]. Most works focus on LiDAR-based 3DOD settings and propose methods to improve performance or efficiency [52, 56] or solve problems that are specific to point clouds, such as KD into sparser point clouds [48, 58]. Some initial works have also proposed concepts for cross-modal KD in 3D semantic segmentation [34] or simple single or stereo camera-based 3DOD models [9, 13, 16, 31, 61]. However, current research focus has shifted to more general multi-camera settings, where up to our knowledge, we are the first to investigate KD across modalities, tasks, and stages comprehensively.

3. Proposed X^{3}KD Framework

We first define our considered problem and baseline in Sec. 3.1. Next, we give an overview on X^{3}KD in Sec. 3.2 presenting specific advancements in Secs. 3.3 and 3.4.

3.1. Problem Formulation and Baseline Method

Problem Definition: We aim at developing a 3DOD model with camera images $\boldsymbol{x} \in \mathbb{R}^{N^{\text{cam}} \times H^{\text{cam}} \times W^{\text{cam}} \times 3}$ as input, where N^{cam}, H^{cam}, and W^{cam} represent the number of images, image height, and image width, respectively, and N^{bbox} 3D bounding boxes $\overline{\boldsymbol{b}} = \left\{ \left(\overline{\boldsymbol{b}}_n^{\text{reg}}, \overline{b}_n^{\text{cls}} \right) n \in \{1, \ldots, N^{\text{bbox}}\} \right\}$ as output. Each bounding box is represented by regression parameters $\overline{\boldsymbol{b}}_n^{\text{reg}} \in \mathbb{R}^9$ (three, three, two, and one for the center, spatial extent, velocity, and yaw angle, respectively), and a classification label $\overline{b}_n^{\text{cls}} \in \mathcal{S}$ from the set of $|\mathcal{S}|$ classes $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$. During training, not only are camera images available, but we can also make use of a 3D LiDAR point cloud $\boldsymbol{l} \in \mathbb{R}^{P \times 5}$ with P points, each one containing the 3D position, intensity, and ring index. The point cloud $\boldsymbol{l}$ is not available during inference.

Baseline Model: We build upon the recently published state-of-the-art method BEVDepth [28], whose setup is depicted in the blue box of Fig. 2. First, all images are processed by a PV feature extractor, yielding features $\boldsymbol{f}^{\text{PV}} \in \mathbb{R}^{N^{\text{cam}} \times H^{\text{PV}} \times W^{\text{PV}} \times C^{\text{PV}}}$ in PV with spatial extent $H^{\text{PV}} \times W^{\text{PV}}$ and number of channels C^{PV}. Afterwards, the features are passed through the Lift-Splat-Shoot transform [40], which predicts discretized depth values $\hat{\boldsymbol{d}}$, transforms pixels corresponding to $\boldsymbol{f}^{\text{PV}}$ into a point cloud representation and obtains BEV features $\boldsymbol{f}^{\text{BEV}} \in \mathbb{R}^{H^{\text{BEV}} \times W^{\text{BEV}} \times C^{\text{BEV}}}$ via voxel pooling. BEV features are further processed by an encoder-decoder network as in [28], yielding refined features $\boldsymbol{f}^{\text{REF}} \in \mathbb{R}^{H^{\text{BEV}} \times W^{\text{BEV}} \times C^{\text{REF}}}$. Finally, the CenterPoint prediction head [55], predicts dense object probability scores $\hat{\boldsymbol{b}}^{\text{cls}} \in \mathbb{I}^{H^{\text{BEV}} \times W^{\text{BEV}} \times |\mathcal{S}|}$ for each class as well as corresponding regression parameters $\hat{\boldsymbol{b}}^{\text{reg}} \in \mathbb{R}^{H^{\text{BEV}} \times W^{\text{BEV}} \times 9}$. The final bounding box predictions $\overline{\boldsymbol{b}}$ are generated by non-learned decoding of these dense representations [55].

Baseline Training: The baseline is trained by optimizing the 3D bounding box losses $\mathcal{L}^{\text{CPoint}}$ from Centerpoint [55] as well as the depth loss $\mathcal{L}^{\text{depth}}$ from [28], yielding

$$\mathcal{L}^{\text{GT}} = \mathcal{L}^{\text{depth}}(\hat{\boldsymbol{d}}, \boldsymbol{d}) + \mathcal{L}^{\text{CPoint}}(\hat{\boldsymbol{b}}^{\text{cls}}, \hat{\boldsymbol{b}}^{\text{reg}}, \boldsymbol{b}), \tag{1}$$

where $\boldsymbol{d}$ is the depth ground truth generated from projected LiDAR points and $\boldsymbol{b}$ is the set of ground truth bounding boxes. For more details, we refer to the supplementary.

3.2. X^{3}KD Overview

Our X^{3}KD framework (Fig. 2) improves the performance of a multi-camera 3DOD model without introducing additional complexity during inference. Hence, our model's inference setup is equal to the one of our baseline. During training, however, we explore multiple knowledge distillation (KD) strategies across modalities, tasks, and stages.

X^{3}KD Loss: First, we employ a pre-trained LiDAR-based 3DOD model, as shown in Fig. 2 (top part). We propose three losses for distilling knowledge across different stages into the camera-based 3DOD: An output-stage distillation (X-OD) loss $\mathcal{L}^{\text{X-OD}}$ between the outputs of the camera and LiDAR models, a feature-stage distillation (X-FD) scheme and a corresponding loss $\mathcal{L}^{\text{X-FD}}$ to guide the focus of the BEV features after the view transformation, and a feature-stage adversarial training (X-AT) with a loss $\mathcal{L}^{\text{X-AT}}$ between the camera and LiDAR model features to encourage their feature similarity. Second, we use an instance segmentation network, cf. Fig. 2 (bottom part). We propose cross-task instance segmentation distillation (X-IS) by imposing a loss $\mathcal{L}^{\text{X-IS}}$ between the output of an additional PV instance segmentation head and teacher-generated pseudo labels. Our total loss for X^{3}KD is then given by:

$$\mathcal{L}^{\text{X}^3\text{KD}} = \sum_{i \in \mathcal{I}} \lambda^i \mathcal{L}^i, \mathcal{I} = \{\text{GT}, \text{X-OD}, \text{X-FD}, \text{X-AT}, \text{X-IS}\} \tag{2}$$

3.3. Cross-modal Knowledge Distillation

The current superiority of LiDAR-based 3DOD over multi-camera 3DOD can be attributed to the ambiguous view transformation in multi-camera models, which may place features at the wrong position in the final representation (*e.g.*, a BEV grid). Meanwhile, LiDAR-based models operate on a 3D point cloud, which can easily be projected onto any view representation. Thereby, the extracted features preserve 3D information. Our cross-modal KD com-

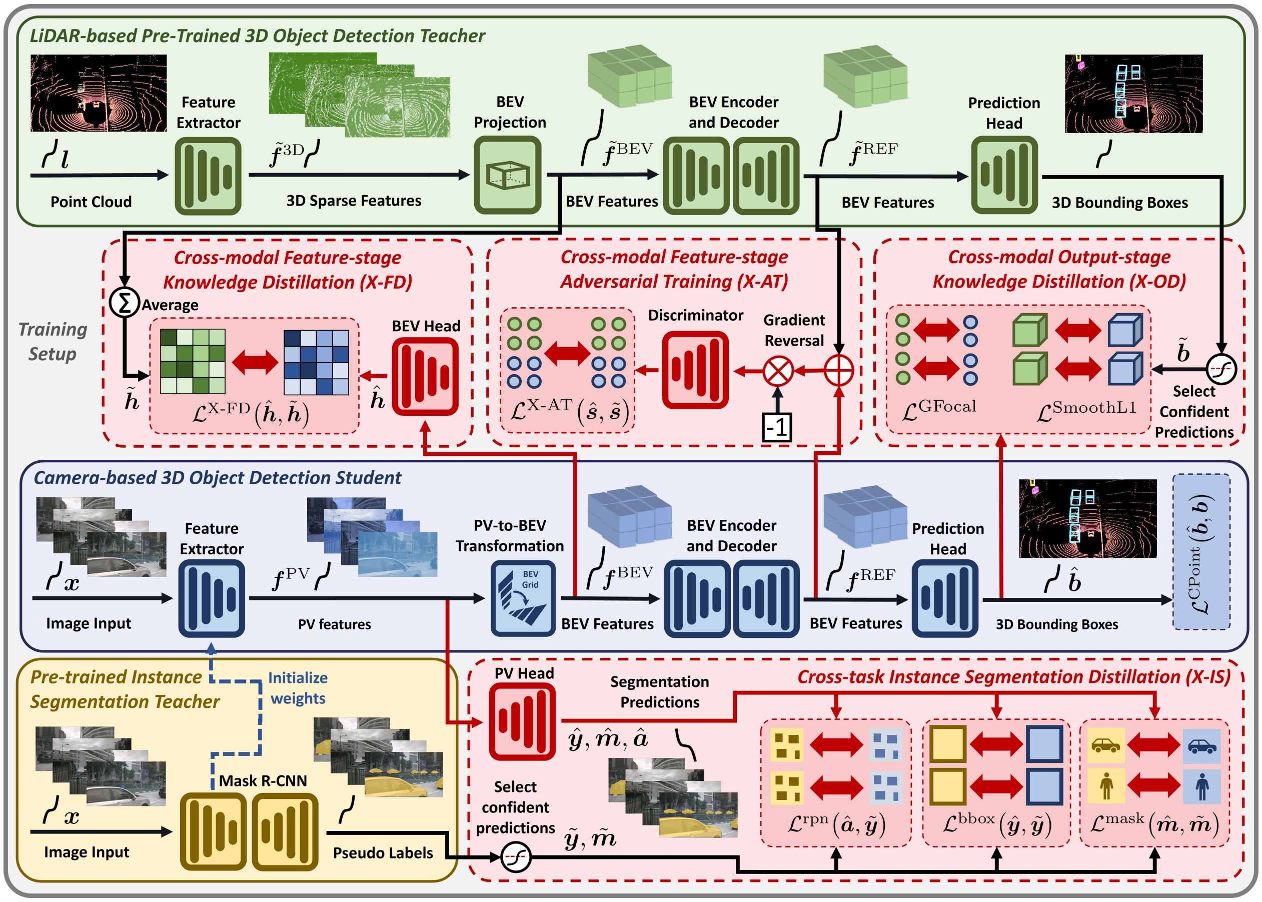

Figure 2. **We present X³KD, a knowledge distillation (KD) framework for multi-camera 3DOD.** We employ an inference setup (middle blue box) relying only on multi-camera image input (LiDAR point cloud in the output is just shown for visualization). During training, we apply KD across several network stages (red arrows originating from the blue box): In perspective-view (PV) feature extraction, we apply cross-task instance segmentation distillation (X-IS) from an instance segmentation teacher (yellow box). In the bird's eye view (BEV), we apply cross-modal feature distillation (X-FD), adversarial training (X-AT), and output distillation (X-OD) from a LiDAR-based 3DOD teacher (green box). X³KD significantly enhances the multi-camera 3DOD without inducing extra complexity during inference.

ponents transfer this knowledge to the multi-camera 3DOD model across different network stages, cf. Fig. 2 (top part).

LiDAR-based 3DOD Model Architecture: Our LiDAR-based 3DOD model is mainly based on CenterPoint [55]. First, the point cloud $\boldsymbol{l} \in \mathbb{R}^{P\times 5}$ is processed by the Sparse Encoder from SECOND [51], yielding 3D sparse features $\tilde{\boldsymbol{f}}^{\text{3D}} \in \mathbb{R}^{H^{\text{BEV}}\times W^{\text{BEV}}\times \tilde{D}^{\text{3D}}\times \tilde{C}^{\text{3D}}}$ with volumetric extent $H^{\text{BEV}} \times W^{\text{BEV}} \times \tilde{D}^{\text{3D}}$ and number of channels $\tilde{C}^{\text{3D}}$. Then, the features are projected onto the same BEV plane as in the camera-based 3DOD model, yielding BEV features $\tilde{\boldsymbol{f}}^{\text{BEV}} \in \mathbb{R}^{H^{\text{BEV}}\times W^{\text{BEV}}\times \tilde{C}^{\text{BEV}}}$ with $\tilde{C}^{\text{BEV}} = \tilde{D}^{\text{3D}} \cdot \tilde{C}^{\text{3D}}$. These are further processed by an encoder-decoder network, yielding refined BEV features $\tilde{\boldsymbol{f}}^{\text{REF}} \in \mathbb{R}^{H^{\text{BEV}}\times W^{\text{BEV}}\times \tilde{C}^{\text{REF}}}$. Finally, the features are passed through a prediction head, yielding probability score maps $\tilde{\boldsymbol{b}}^{\text{cls}} \in \mathbb{I}^{H^{\text{BEV}}\times W^{\text{BEV}}\times |\mathcal{S}|}$ and regression maps $\tilde{\boldsymbol{b}}^{\text{reg}} \in \mathbb{R}^{H^{\text{BEV}}\times W^{\text{BEV}}\times 9}$ analogous to the outputs $\hat{\boldsymbol{b}}^{\text{cls}}$ and $\hat{\boldsymbol{b}}^{\text{reg}}$ of the multi-camera 3DOD model.

Output-stage Distillation (X-OD): Following many approaches in KD [6, 11, 15, 54, 57], we distill knowledge at the output stage by imposing losses between the teacher's outputs $\tilde{\boldsymbol{b}}^{\text{cls}}$ and $\tilde{\boldsymbol{b}}^{\text{reg}}$ and the student's outputs $\hat{\boldsymbol{b}}^{\text{cls}}$ and $\hat{\boldsymbol{b}}^{\text{reg}}$. Specifically, we impose a Gaussian focal loss $\mathcal{L}^{\text{GFocal}}$ [26] between $\hat{\boldsymbol{b}}^{\text{cls}}$ and $\tilde{\boldsymbol{b}}^{\text{cls}}$ to put more weight on rare classes and compensate for the class imbalance. As this loss only considers pseudo labels as a positive sample if they are exactly 1, we select high-confidence teacher output probabilities $\tilde{\boldsymbol{b}}^{\text{cls}}$, *i.e.*, probability values over a threshold $\alpha^{\text{3D-bbox}}$, and set them to 1. Further, the regression output of the student $\hat{\boldsymbol{b}}^{\text{reg}}$ is supervised by the corresponding output $\tilde{\boldsymbol{b}}^{\text{reg}}$ of the teacher by imposing a Smooth L1 loss $\mathcal{L}^{\text{SmoothL1}}$ [12]. Finally, we propose to weigh the regression loss by the teacher's pixel-wise averaged output probabilities $\langle \tilde{\boldsymbol{b}}^{\text{cls}}_s \rangle = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \tilde{\boldsymbol{b}}^{\text{cls}}_s \in \mathbb{R}^{H^{\text{BEV}}\times W^{\text{BEV}}}$ to weigh regions which likely contain objects higher than the background. Overall, X-OD is defined as:

$$\mathcal{L}^{\text{X-OD}}\left(\hat{\boldsymbol{b}}, \tilde{\boldsymbol{b}}\right) = \mathcal{L}^{\text{GFocal}}\left(\hat{\boldsymbol{b}}^{\text{cls}}, \tilde{\boldsymbol{b}}^{\text{cls}}\right) + \mathcal{L}^{\text{SmoothL1}}\left(\hat{\boldsymbol{b}}^{\text{reg}}, \tilde{\boldsymbol{b}}^{\text{reg}}\right) \quad (3)$$

Feature-stage Distillation (X-FD): Our X-FD compo-

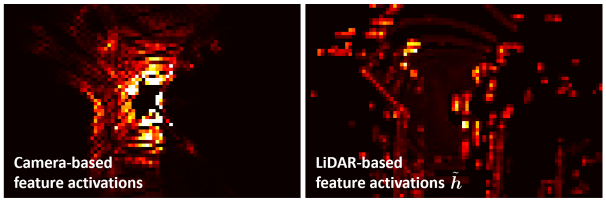

Figure 3. **Mean feature activations** from the camera-based student after the view transformation (left) and the LiDAR-based teacher (right) exhibit structural dissimilarity.

nent exploits the precise and sparse nature of features extracted from LiDAR point clouds, which precisely encode locations of relevant objects for 3DOD. Thereby, the mean sparse feature activation $\tilde{\boldsymbol{h}}$, cf. Fig. 3 (right), provides a good initial estimate for the potential location of objects. While it would be natural to impose similarity losses between BEV features from the camera and LiDAR models, these features are structurally quite different (cf. Fig. 3), such that our attempts to impose such losses did lead to unstable training behavior. Therefore, we add a small BEV decoder to the multi-camera model, which outputs a prediction $\hat{\boldsymbol{h}}$ for the mean sparse feature activations from the LiDAR teacher $\tilde{\boldsymbol{h}}$. The X-FD loss $\mathcal{L}^{\text{X-FD}}$ is then given as:

$$\mathcal{L}^{\text{X-FD}} = \text{L1}\big(\hat{\boldsymbol{h}}, \tilde{\boldsymbol{h}}\big) \tag{4}$$

Feature-stage Adversarial Training (X-AT): We further propose X-AT to encourage a more global feature similarity between the refined features $\boldsymbol{f}^{\text{REF}}$ and $\tilde{\boldsymbol{f}}^{\text{REF}}$ from both modalities in BEV space. Due to the structural dissimilarity of features from both modalities directly after the BEV projection (Fig. 3), we apply the adversarial training on the refined features $\tilde{\boldsymbol{f}}^{\text{REF}}$ and $\hat{\boldsymbol{f}}^{\text{REF}}$. We pass these cross-modal features through a gradient reversal layer and a patch-based discriminator network [20], which outputs two modality-specific probabilities. The discriminator is optimized to classify the features by modality using a binary cross-entropy loss $\mathcal{L}^{\text{X-AT}}$ between the output probabilities $\hat{\boldsymbol{s}}$ and the ground truth modality labels $\boldsymbol{s}$:

$$\mathcal{L}^{\text{X-AT}} = \text{BCE}\left(\hat{\boldsymbol{s}}, \boldsymbol{s}\right) \tag{5}$$

We then encourage modality-agnostic features in the multi-camera 3DOD model through gradient reversal.

3.4. Cross-task Knowledge Distillation

Learning a good feature representation in PV is difficult when all supervision signals are backpropagated through an ambiguous view transformation. As a possible solution, M^{2}BEV [49] proposes instance segmentation (IS) pretraining. However, deep neural networks exhibit catastrophic forgetting such that this initial knowledge is not necessarily preserved during 3DOD training. Therefore, we propose cross-task instance segmentation distillation (X-IS) to preserve the knowledge contained in the PV features continuously. Specifically, we use the outputs of a pre-trained instance segmentation network as pseudo labels to optimize an additional PV instance segmentation head, cf. Fig. 2.

Pseudo Label Generation: In this work, we use the well-established Mask R-CNN architecture [14] as a teacher; see Fig. 2 (bottom left). We use its original architecture, consisting of a feature extractor, a feature pyramid network (FPN), a region proposal network (RPN), and a region of interest (ROI) head, including a mask branch. As output, we obtain N^{IS} bounding boxes $\tilde{\boldsymbol{y}} = \big\{\big(\tilde{\boldsymbol{y}}_n^{\text{bbox}}, \tilde{y}_n^{\text{cls}}, \tilde{y}_n^{\text{score}}\big), n \in \big\{1, \ldots, N^{\text{IS}}\big\}\big\}$ with four parameters for bounding box center and spatial extent $\tilde{\boldsymbol{y}}_n^{\text{bbox}} \in \mathbb{R}^4$, a classification result $\tilde{y}_n^{\text{cls}} \in \mathcal{S}^{\text{IS}}$ from the set of IS classes $\mathcal{S}^{\text{IS}}$, and an objectness score $\tilde{y}_n^{\text{score}} \in \mathbb{I}$ with $\mathbb{I} = [0, 1]$. Additionally, we obtain corresponding object masks $\tilde{\boldsymbol{m}} = \big\{\tilde{\boldsymbol{m}}_n, n \in \big\{1, \ldots, N^{\text{IS}}\big\}\big\}$ with single masks $\tilde{\boldsymbol{m}}_n \in \{0, 1\}^{H_n^{\text{mask}} \times W_n^{\text{mask}}}$ and spatial resolution $H_n^{\text{mask}} \times W_n^{\text{mask}}$. We select all samples with a score $\tilde{y}_n^{\text{score}} > \alpha^{\text{2D-bbox}}$ as pseudo labels.

X-IS Loss Computation: The teacher-generated pseudo labels are used to supervise an additional PV instance segmentation head, cf Fig. 2 (bottom right), which uses the same RPN and ROI head architectures as the teacher. The RPN head outputs region proposals $\hat{\boldsymbol{a}} = \big(\hat{\boldsymbol{a}}^{\text{cls}}, \hat{\boldsymbol{a}}^{\text{reg}}\big)$ with foreground/background scores $\hat{\boldsymbol{a}}^{\text{cls}} \in \mathbb{I}^{H^{\text{PV}} \times W^{\text{PV}} \times 2K}$ and regression parameters $\hat{\boldsymbol{a}}^{\text{reg}} \in \mathbb{R}^{H^{\text{PV}} \times W^{\text{PV}} \times 4K}$ relative to each of the K anchors. Our RPN loss $\mathcal{L}^{\text{rpn}}$ is then comprised of an assignment strategy between pseudo GT and PV head outputs as detailed in [42] and subsequent application of BCE and L1 differences for optimizing $\hat{\boldsymbol{a}}^{\text{cls}}$ and $\hat{\boldsymbol{a}}^{\text{reg}}$, respectively. The N^{RPN} region proposals with the highest foreground scores are subsequently passed through the ROI head, which outputs refined bounding boxes $\hat{\boldsymbol{y}} = \big\{\big(\hat{\boldsymbol{y}}_n^{\text{bbox}}, \hat{\boldsymbol{y}}_n^{\text{cls}}\big), n \in \big\{1, \ldots, N^{\text{RPN}}\big\}\big\}$ with class probabilities $\hat{\boldsymbol{y}}_n^{\text{cls}} \in \mathbb{I}^{|\mathcal{S}^{\text{IS}}|}$, four bounding box regression parameters $\hat{\boldsymbol{y}}_n^{\text{bbox}} \in \mathbb{R}^4$ as well as class-specific mask probabilities $\hat{\boldsymbol{m}} = \big\{\hat{\boldsymbol{m}}_n, n \in \big\{1, \ldots, N^{\text{IS}}\big\}\big\}$ with single masks $\hat{\boldsymbol{m}}_n \in \mathbb{I}^{H^{\text{mask}} \times W^{\text{mask}} \times |\mathcal{S}^{\text{IS}}|}$. Our bounding box loss $\mathcal{L}^{\text{bbox}}$ is comprised of an assignment strategy between ground truth $\tilde{\boldsymbol{y}}$ and prediction $\hat{\boldsymbol{y}}$ and subsequent application of L1 difference between $\hat{\boldsymbol{y}}_n^{\text{bbox}}$ and $\tilde{\boldsymbol{y}}_n^{\text{bbox}}$ as well as cross-entropy (CE) difference between $\hat{\boldsymbol{y}}_n^{\text{cls}}$ and one-hot encoded $\tilde{y}_n^{\text{cls}}$. For computing the mask loss $\mathcal{L}^{\text{mask}}$, we apply a binary cross entropy (BCE) difference between ground truth $\tilde{\boldsymbol{m}}$ and prediction $\hat{\boldsymbol{m}}$, selecting only the output corresponding to the ground truth mask's class. More details can be found in [14]. Overall, our X-IS loss $\mathcal{L}^{\text{X-IS}}$ can be written as:

$$\mathcal{L}^{\text{X-IS}} = \mathcal{L}^{\text{rpn}}\left(\hat{\boldsymbol{a}}, \tilde{\boldsymbol{y}}\right) + \mathcal{L}^{\text{bbox}}\left(\hat{\boldsymbol{y}}, \tilde{\boldsymbol{y}}\right) + \mathcal{L}^{\text{mask}}\left(\hat{\boldsymbol{m}}, \tilde{\boldsymbol{m}}\right). \tag{6}$$

4. Experiments

We first provide our experimental setup (Sec. 4.1) and a state-of-the-art comparison (Sec. 4.2). Next, we verify and analyze our method's components in Secs. 4.3 and 4.4. Last, we evaluate RADAR-based models (Sec. 4.5).

Set	*Model*	*Backbone*	*Resolution*	*mATE*↓	*mASE*↓	*mAOE*↓	*mAVE*↓	*mAAE*↓	*mAP*↑	*NDS*↑
Validation	BEVDet [18]	ResNet-50	256 × 704	0.725	0.279	0.589	0.860	0.245	29.8	37.9
	BEVDet4D [49]			0.703	0.278	0.495	0.354	0.206	32.2	45.7
	BEVDepth [28]			0.629	**0.267**	0.479	0.428	0.198	35.1	47.5
	BEVDepth†			0.636	0.272	0.493	0.499	0.198	35.9	47.2
	STS* [47]			0.601	0.275	0.450	0.446	0.212	37.7	48.9
	BEVStereo* [27]			**0.598**	0.270	**0.438**	0.367	**0.190**	37.2	50.0
	X^{3}KD$_{\text{all}}$	ResNet-50	256 × 704	0.615	0.269	0.471	**0.345**	0.203	**39.0**	**50.5**
Validation	PETR [32]	ResNet-101	512 × 1408	0.710	0.270	0.490	0.885	0.224	35.7	42.1
	BEVDepth†			0.579	0.265	0.387	0.364	0.194	40.9	53.1
	BEVDepth [28]			0.565	0.266	0.358	0.331	**0.190**	41.2	53.5
	STS* [47]			**0.525**	0.262	0.380	0.369	0.204	43.1	54.2
	X^{3}KD$_{\text{all}}$	ResNet-101	512 × 1408	0.552	**0.257**	**0.338**	**0.328**	0.199	**44.8**	**55.3**
Validation	DETR3D [46]	ResNet-101	900 × 1600	0.716	0.268	0.379	0.842	0.200	34.9	43.4
	BEVFormer [30]			0.673	0.274	0.372	0.394	0.198	41.6	51.7
	PolarFormer [21]			0.648	0.270	0.348	0.409	0.201	43.2	52.8
	BEVDepth†	ResNet-101	640 × 1600	0.571	0.260	0.379	0.374	**0.196**	42.8	53.6
	X^{3}KD$_{\text{all}}$	ResNet-101	640 × 1600	**0.539**	**0.255**	**0.320**	**0.324**	**0.196**	**46.1**	**56.7**
Test	BEVFormer [30]	ResNet-101	640 × 1600	0.631	0.257	0.405	0.435	0.143	44.5	53.5
	BEVDepth†			0.533	0.254	0.443	0.404	**0.129**	43.1	53.9
	PolarFormer [21]			0.610	0.258	**0.391**	0.458	**0.129**	**45.6**	54.3
	X^{3}KD$_{\text{all}}$	ResNet-101	640 × 1600	**0.506**	**0.253**	0.414	**0.366**	0.131	**45.6**	**56.1**

Table 2. **Performance comparison on the nuScenes dataset**: We ensure comparability regarding backbone and image resolution. Baseline results are cited except for BEVDepth† which we reproduced in our framework; * indicates recent ArXiv works; best numbers in boldface.

Model	*LET-3D-AP*↑ *All*	*LET-3D-APL*↑ *Vehicle*	*Pedestrian*	*Cyclist*	*All*
BEVDepth†	38.1	40.9	24.1	15.0	26.7
X^{3}KD$_{\text{modal}}$	39.1	41.9	24.6	17.3	27.9
X^{3}KD$_{\text{all}}$	**39.6**	**43.4**	**25.4**	**17.7**	**28.8**

Table 3. **Performance comparison on the Waymo dataset.** We compare X^{3}KD to our re-implemented baseline BEVDepth† [28].

4.1. Experimental Setup

X^{3}KD is implemented using mmdetection3d [10] and PyTorch [39] libraries and trained on 4 NVIDIA A100 GPUs.[1] Here, we describe our main setup on nuScenes while more details are provided in the supplementary.

Datasets: Similar to most recent works [1, 27, 28, 30, 55], we evaluate on the nuScenes and Waymo benchmark datasets. The nuScenes dataset [5] contains 28K, 6K, and 6K samples for training, validation, and test, respectively. We use data from a LiDAR sensor and 6 cameras with bounding box annotations for 10 classes. For the Waymo dataset [43], we use the data from a LiDAR sensor and 5 cameras with annotations for cars, pedestrians, and cyclists. It provides 230K annotated frames from 798, 202, and 150 sequences for training, validation, and test, respectively.

Evaluation Metrics: For nuScenes, we employ the officially defined *mAP* and *NDS* metrics. The *NDS* metric considers *mAP* as well as true positive (*TP*) metrics $\mathbb{TP} = \{mATE, mASE, mAOE, mAVE, mAAE\}$ for translation, scale, orientation, velocity, and attribute, respectively, *i.e.*, $NDS = \frac{1}{10}\left(5 \cdot mAP\right) + \sum_{TP \in \mathbb{TP}} 1 - \min\left(1, TP\right)$. For Waymo, we employ the official metrics of the camera-only 3D object detection track [19]: The *LET-3D-AP* calculates average precision after longitudinal error correction, while *LET-3D-APL* also penalizes the longitudinal error.

Network Architecture and Training: For a fair comparison, our network architecture follows previous works [17, 21, 27, 28, 30, 47]. We consider the ResNet-50-based setting with a resolution of 256 × 704 and the ResNet-101-based setting with resolutions of 512 × 1408 or 640 × 1600. Further network design choices are adopted from [17]. We train all models for 24 epochs using the CBGS training strategy [62], a batch size of 16 and AdamW [36] with an initial learning rate of $2 \cdot 10^{-4}$. The loss weights are set to $\lambda^{\text{GT}} = 1$, $\lambda^{\text{X-FD}} = 10$, $\lambda^{\text{X-AT}} = 10$, $\lambda^{\text{X-OD}} = 1$, and $\lambda^{\text{X-IS}} = 1$ while the thresholds are set to $\alpha^{\text{3D-bbox}} = 0.6$ and $\alpha^{\text{2D-bbox}} = 0.2$. Our LiDAR teacher is based on the CenterPoint architecture [55] and the TransFusion training schedule [1]. The supplementary contains further explanations, hyperparameter studies, and configurations for the Waymo dataset.

4.2. State-of-the-art Comparisons

We perform a comparison of X^{3}KD with all contributions, *i.e.*, X^{3}KD$_{\text{all}}$, to other SOTA methods in Table 2. In the ResNet-50-based setting, our model achieves the best results with scores of 39.0 and 50.5 in *mAP* and *NDS*, respectively. In the high-resolution ResNet-101-based setting, our model achieves SOTA scores of 46.1 and 56.7. *At this resolution, we outperform all previous SOTA methods in all considered metrics and outperform the second best result by 2.9 points in mAP and 2.5 points in NDS.* To explicitly show that our method improves on top of current SOTA baselines, we retrain our strongest baseline among

[1]We use mmdetection3d v1.0, Python 3.8, PyTorch 1.11, CUDA 11.3

Model	X-OD	X-FD	X-AT	X-IS	*mATE*↓	*mASE*↓	*mAOE*↓	*mAVE*↓	*mAAE*↓	*mAP*↑	*NDS*↑
BEVDepth†	✗	✗	✗	✗	0.636	0.272	0.493	0.499	0.198	35.9	47.2
X-OD	✓	✗	✗	✗	0.642	0.278	**0.456**	**0.338**	**0.188**	35.7	48.7
X-FD	✗	✓	✗	✗	0.644	0.276	0.479	0.361	0.200	36.1	48.5
X-AT	✗	✗	✓	✗	0.648	0.277	0.492	0.354	<u>0.192</u>	35.5	48.1
X^3KD_{modal}	✓	✓	✓	✗	<u>0.632</u>	<u>0.271</u>	**0.456**	<u>0.342</u>	0.203	36.8	49.4
X-IS	✗	✗	✗	✓	0.635	0.273	<u>0.462</u>	0.350	0.204	<u>38.7</u>	<u>50.1</u>
X^3KD_{all}	✓	✓	✓	✓	**0.615**	**0.269**	0.471	0.345	0.203	**39.0**	**50.5**
LiDAR Teacher	NA	NA	NA	NA	0.301	0.257	0.298	0.256	0.195	59.0	66.4

Table 4. **Ablation study of X^3KD on the nuScenes validation set**: We incrementally add our proposed cross-modal feature distillation (X-FD), adversarial training (X-AT) and output distillation (X-OD) as well as our cross-task instance segmentation distillation (X-IS). All X^3KD variants in the top part are solely based on multi-camera images during inference. Best numbers in boldface, second best underlined.

Model	*Dist.*	*Weight*	*w/o GT*	*mAOE*↓	*mAVE*↓	*mAP*↑	*NDS*↑
BEVDepth†	✗	✗	✗	0.493	0.499	**35.9**	47.2
	✓	✗	✗	0.477	0.342	35.6	48.5
X-OD	✓	✓	✗	**0.456**	**0.338**	35.7	**48.7**
	✓	✗	✓	1.090	0.972	36.1	35.3
$X\text{-}OD_{w/o\ GT}$	✓	✓	✓	**0.724**	**0.570**	**36.5**	**43.7**

Table 5. **Ablation study on cross-modal output distillation (X-OD) on the nuScenes validation set.** We show the effect of weighing the regression loss in (3) by the teacher output probabilities $\langle\tilde{\boldsymbol{b}}_s^{\mathrm{cls}}\rangle$ (Weight) during distillation (Dist.). We also show that our method can be trained without annotations (w/o GT).

Model	*Student Backbone*	*Teacher Backbone*	*Pre.*	*Dist.*	*mAP*↑	*NDS*↑
BEVDepth†	ResNet-50	NA	✗	✗	35.9	47.2
	ResNet-50	ResNet-50	✗	✓	36.4	48.8
	ResNet-50	NA	✓	✗	37.7	49.5
X-IS	ResNet-50	ResNet-50	✓	✓	**38.7**	**50.1**
X-IS	ResNet-50	ConvNeXt-T	✓	✓	38.5	49.9
	ConvNeXt-T	NA	✗	✗	38.3	50.8
	ConvNeXt-T	ResNet-50	✗	✓	**38.9**	**51.4**

Table 6. **Ablation study on cross-task instance segmentation distillation (X-IS) on the nuScenes validation set.** We evaluate the effect of using pre-trained weights (Pre.) and knowledge distillation (Dist.) as well as different teacher/student backbones.

published works, *i.e.*, BEVDepth [28], in our code framework, dubbed BEVDepth†. At all resolutions, we are able to closely reproduce the reported results and improve by about 3 points in both *mAP* and *NDS* upon them. *On the test set, we outperform the second best approach PolarFormer [21] by* 1.8 *points in terms of the main NDS metric* and achieve best results in 5 out of 7 metrics. We also show results for BEVDepth† and X^3KD variants on the Waymo dataset in Table 3. As on nuScenes, our X^3KD_{all} model clearly outperforms the baseline in all metrics.

4.3. Method Ablation Studies

Effectiveness of the Proposed Components: We incrementally add our contributions in Table 4 and evaluate them in terms of *NDS* and *mAP*. First, we individually add X-OD, X-FD, and X-AT. For all three components, there is an improvement in the *NDS* metric, while the *mAP* metric remains similar or slightly worse. Adding all three components (X^3KD_{modal}) gives a clear improvement over the baseline as well as applying each component individually. Particularly, we observe that the *additional cross-modal supervision improves bounding box velocity estimation from multi-camera input* as can be seen by the apparent improvement in the *mAVE* metric. Using X-IS, surprisingly gives an even more substantial improvement. This might indicate that *supervision in BEV cannot completely compensate for the lack of rich features in PV*. Finally, adding all components together to *our proposed* X^3KD_{all} *model clearly outperforms all other variants* in terms of the main *NDS* and *mAP* metrics and is best in 4 out of 7 metrics in Table 4.

Cross-Modal Output Distillation (X-OD): We provide insights into our X-OD design in Table 5. In the top part, we observe that models trained with output distillation improve over the baseline in terms of *NDS* and that the confidence-based weighting is particularly effective for orientation (*mAOE*) and velocity (*mAVE*) prediction. Further, we train the multi-camera 3DOD without using annotations (Table 5, bottom part) solely from KD. In this setting, the weighting yields even more significant improvements in particular in terms of the *NDS* metric. Also, *the* $X\text{-}OD_{w/o\ GT}$ *model surprisingly outperforms the model variants trained with annotations in terms of the mAP metric.* This promising result indicates that future work might be able to use large-scale pre-training with KD on unlabelled data for further performance improvements.

Cross-task Instance Segmentation Distillation (X-IS): Ablations on our X-IS design are shown in Table 6. We observe that initialization of the backbone with weights from a pre-trained instance segmentation as well as cross-task distillation, improves the baseline's result. Combining both aspects to X-IS yields the best result in both *mAP* and *NDS*. Using a different teacher model based on ConvNeXt-T yields similarly good results and shows that the feature extraction architectures of the instance segmentation teacher and the multi-camera 3DOD student do not need to match. Also, knowledge can be distilled from a simple ResNet-50-based model into a more sophisticated architecture such as ConvNeXt-T (bottom part of Table 6). Overall, *cross-task*

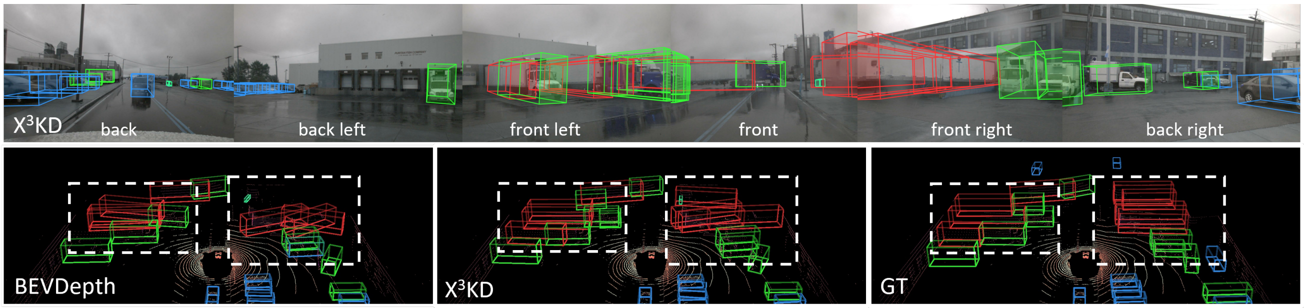

Figure 4. **Qualitative results on nuScenes**: We show the multi-camera input (top) and bounding box visualizations (bottom). We compare ResNet-101-based X^{3}KD$_{\text{all}}$ to BEVDepth† and the ground truth (GT) for a resolution of 640 $\times$ 1600. Best viewed on screen and in color.

Model	*RADAR Input*	*Cam. Input*	Validation *mAP*↑	Validation *NDS*↑	Test *mAP*↑	Test *NDS*↑
RADAR only	✓	✗	12.9	13.0	-	-
X^{3}KD$_{\text{modal}}$	✓	✗	**17.7**	**23.5**	-	-
Fusion only	✓	✓	38.9	51.0	40.2	52.3
X^{3}KD$_{\text{modal}}$	✓	✓	**42.3**	**53.8**	**44.1**	**55.3**

Table 7. **Generalization of our method to RADAR:** We distill knowledge from a LiDAR-based 3DOD into a RADAR-based and a RADAR-camera fusion-based 3DOD model. For RADAR-based models, we report the mAP just for the car class as these models underperform on other classes due to the point cloud sparsity.

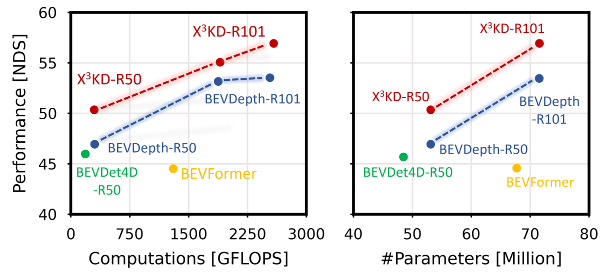

Figure 5. **Complexity Analysis** of X^{3}KD in comparison to BEVDepth [28], BEVDet4D [17], and BEVFormer [30].

distillation can improve performance without requiring an additional pre-training step.

4.4. Method Analysis

Performance-Complexity Trade-off: We analyze our method's efficiency compared to state-of-the-art methods [17, 28, 30] in Fig. 5. We compare to reimplementations of BEVDepth [28] and BEVDet4D [17] as well as reported results of BEVFormer [30]. All reported models are ResNet-50-based or ResNet-101-based to ensure that a better trade-off cannot be attributed to a more efficient backbone. We observe that X^{3}KD (red curve) outperforms BEVDepth (blue curve) at equal complexity due to the improved supervision from KD. Also, compared to BEVDet4D and BEVFormer a better trade-off can be observed, likely because of the absence of LiDAR supervision in BEVDet4D and the complex Transformer model in BEVFormer. Accordingly, *our results show that X^{3}KD achieves a better complexity-performance trade-off than current state-of-the-art methods.*

Qualitative Results: We further show qualitative results of X^{3}KD and BEVDepth in Fig. 4. As highlighted by the white boxes, X^{3}KD detects and places objects more accurately in the scene. In particular, the recognition of objects and the prediction of their orientation shows improved characteristics in the X^{3}KD output, which is coherent with a better quantitative performance of X^{3}KD in Table 4. Further qualitative results are given in the supplementary.

4.5. Generalization to RADAR

We also generalize X^{3}KD to RADAR-based and camera-RADAR fusion-based models. For RADAR-based models, we cannot apply cross-task KD from the instance segmentation teacher. Hence, we only use the cross-modal KD contributions, *i.e.*, X^{3}KD$_{\text{modal}}$. Our results on the nuScenes validation set show that X^{3}KD$_{\text{modal}}$ significantly enhances the performance in both settings. Notably, the transfer from camera to RADAR was straightforward as we achieved the reported improvements without requiring tuning of hyperparameters. Further, we evaluate our fusion-based X^{3}KD$_{\text{modal}}$ model on the nuScenes test set, where *we outperform all other Camera-RADAR, fusion-based models, hence setting the state-of-the-art result.*

5. Conclusions

We proposed X^{3}KD, a KD framework for multi-camera 3DOD. By distilling across tasks from an instance segmentation teacher and across modalities from a LiDAR-based 3DOD teacher into different stages of a multi-camera 3DOD student, we show that the model performance can be enhanced without inducing additional complexity during inference. We evaluated X^{3}KD on the nuScenes and Waymo datasets, outperforming previous approaches by 2.9% *mAP* and 2.5% *NDS*. The transferability to other sensors, such as RADAR, and the possibility to train 3DOD models without annotations further demonstrate X^{3}KD's effectiveness. Combining these two findings could be used in future applications to train 3DOD models for arbitrary sensors, requiring only a LiDAR-based 3DOD model.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection With Transformers. In *Proc. of CVPR*, pages 1090–1099, 2022. 1, 2, 3, 6

[2] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-Align: Cross-Modal Cross-View Alignment for Bird's-Eye-View Segmentation. In *Proc. of WACV*, pages 3287–3297, 2023. 2

[3] Shubhankar Borse, Hyojin Park, Hong Cai, Debasmit Das, Risheek Garrepalli, and Fatih Porikli. Panoptic, Instance and Semantic Relations: A Relational Context Encoder to Enhance Panoptic Segmentation. In *Proc. of CVPR*, pages 1269–1279, 2022. 2

[4] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. A Loss Function for Structured Boundary-Aware Segmentation. In *Proc. of CVPR*, pages 5901–5911, 2021. 2

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proc. of CVPR*, pages 11621–11631, 2020. 6

[6] Hong Cai, Janarbek Matai, Shubhankar Borse, Yizhe Zhang, Amin Ansari, and Fatih Porikli. X-Distill: Improving Self-Supervised Monocular Depth via Cross-Task Distillation. In *Proc. of BMVC*, pages 1–14, 2021. 4

[7] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. FUTR3D: A Unified Sensor Fusion Framework for 3D Detection. *arXiv preprint arXiv:2203.10642*, 2022. 3

[8] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinghong Jiang, Feng Zhao, Bolei Zhou, and Hang Zhao. AutoAlign: Pixel-Instance Feature Aggregation for Multi-Modal 3D Object Detection. In *Proc. of IJCAI*, pages 1–7, 2022. 2

[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. In *Proc. of ICLR*, pages 1–17, 2022. 3

[10] MMDetection3D Contributors. MMDetection3D: Open-MMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020. 6

[11] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General Instance Distillation for Object Detection. In *Proc. of CVPR*, pages 7842–7851, 2021. 4

[12] Ross Girshick. Fast R-CNN. In *Proc. of ICCV*, pages 1440–1448, 2015. 4

[13] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. LIGA-Stereo: Learning LiDAR Geometry Aware Representations for Stereo-Based 3D Detector. In *Proc. of ICCV*, pages 3153–3163, 2021. 3

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. of CVPR*, pages 2961–2969, 2017. 5

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. In *Proc. of NIPS*, pages 1–9, 2014. 3, 4

[16] Yu Hong, Hang Dai, and Yong Ding. Cross-Modality Knowledge Distillation Network for Monocular 3D Object Detection. In *Proc. of ECCV*, pages 87–104, 2022. 3

[17] Junjie Huang and Guan Huang. BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection. *arXiv preprint arXiv:2203.17054*, 2022. 2, 6, 8

[18] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View. *arXiv preprint arXiv:2112.11790*, 2021. 2, 6

[19] Wei-Chih Hung, Henrik Kretzschmar, Vincent Casser, Jyh-Jing Hwang, and Dragomir Anguelov. LET-3D-AP: Longitudinal Error Tolerant 3D Average Precision for Camera-Only 3D Detection. *arXiv preprint arXiv:2206.07705*, 2022. 6

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-To-Image Translation With Conditional Adversarial Networks. In *Proc. of CVPR*, pages 1125–1134, 2017. 5

[21] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. PolarFormer: Multi-camera 3D Object Detection with Polar Transformers. In *Proc. of AAAI*, pages 1–9, 2023. 6, 7

[22] Yang Jiao, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Xiaolin Wei, Lin Ma, and Yu-Gang Jiang. MSMDFusion: Fusing LiDAR and Camera at Multiple Scales with Multi-Depth Seeds for 3D Object Detection. *arXiv preprint arXiv:2209.03102*, 2022. 2

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumarana, and R. Hadsella. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2016. 2

[24] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3D Proposal Generation and Object Detection from View Aggregation. In *Proc. of IROS*, pages 1–8, 2018. 3

[25] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proc. of CVPR*, pages 12697–12705, 2019. 1

[26] Hei Law and Jia Deng. CornerNet: Detecting Objects as Paired Keypoints. In *Proc. of ECCV*, pages 734–750, 2018. 4

[27] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. BEVStereo: Enhancing Depth Estimation in Multi-view 3D Object Detection with Dynamic Temporal Stereo. *arXiv preprint arXiv:2209.10248*, 2022. 2, 6

[28] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. BEVDepth:

Acquisition of Reliable Depth for Multi-view 3D Object Detection. In *Proc. of AAAI*, pages 1–9, 2023. 1, 2, 3, 6, 7, 8

[29] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V. Le, Alan Yuille, and Mingxing Tan. DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection. In *Proc. of CVPR*, pages 17182–17191, 2022. 2

[30] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. In *Proc. of ECCV*, pages 1–20, 2022. 1, 2, 6, 8

[31] Junjie Liu and Weiyu Yu. Multi-view LiDAR Guided Monocular 3D Object Detection. In *Proc. of PRCV*, pages 520–532, 2022. 3

[32] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position Embedding Transformation for Multi-View 3D Object Detection. In *Proc. of ECCV*, pages 1–18, 2022. 2, 6

[33] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images. *arXiv preprint arXiv:2206.01256*, 2022. 2

[34] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 3D-to-2D Distillation for Indoor Scene Parsing. In *Proc. of CVPR*, pages 4464–4474, 2021. 3

[35] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. *arXiv preprint arXiv:2205.13542*, 2022. 2

[36] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proc. of ICLR*, pages 1–18, 2018. 6

[37] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is Pseudo-Lidar Needed for Monocular 3D Object Detection? In *Proc. of CVPR*, pages 3142–3152, 2021. 1

[38] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time Will Tell: New Outlooks and A Baseline for Temporal Multi-View 3D Object Detection. *arXiv preprint arXiv:2210.02443*, 2022. 2

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. of NeurIPS*, pages 8024–8035, 2019. 6

[40] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *Proc. of ECCV*, pages 194–210, 2020. 1, 2, 3

[41] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical Depth Distribution Network for Monocular 3D Object Detection. In *Proc. of CVPR*, pages 8555–8564, 2021. 1

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 5

[43] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, and Aleksei Timofeev. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proc. of CVPR*, pages 2446–2454, 2020. 6

[44] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential Fusion for 3D Object Detection. In *Proc. of CVPR*, pages 4604–4612, 2020. 2

[45] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proc. of CVPR*, pages 11794–11803, 2021. 2

[46] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. In *Proc. of CoRL*, pages 180–191, 2022. 1, 2, 6

[47] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. STS: Surround-view Temporal Stereo for Multi-view 3D Detection. *arXiv preprint arXiv:2208.10145*, 2022. 2, 6

[48] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. LiDAR Distillation: Bridging the Beam-Induced Domain Gap for 3D Object Detection. In *Proc. of ECCV*, pages 1–18, 2022. 3

[49] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M. Alvarez. M^2BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. *arXiv preprint arXiv:2204.05088*, 2022. 2, 5, 6

[50] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. FusionPainting: Multimodal Fusion with Adaptive Attention for 3D Object Detection. In *Proc. of ITSC*, pages 3047–3054, 2021. 2

[51] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, 2018. 4

[52] Jihan Yang, Shaoshuai Shi, Runyu Ding, Zhe Wang, and Xiaojuan Qi. Towards Efficient 3D Object Detection with Knowledge Distillation. In *Proc. of NeurIPS*, pages 3289–3298, 2021. 3

[53] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. DeepInteraction: 3D Object Detection via Modality Interaction. In *Proc. of NeurIPS*, pages 1–13, 2022. 2

[54] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and Global Knowledge Distillation for Detectors. In *Proc. of CVPR*, pages 4643–4652, 2022. 4

[55] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. In *Proc. of CVPR*, pages 11784–11793, 2021. 1, 3, 4, 6

[56] 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. 3

[57] Yizhe Zhang, Shubhankar Borse, Hong Cai, and Fatih Porikli. AuxAdapt: Stable and Efficient Test-Time Adaptation for Temporally Consistent Video Semantic Segmentation. In *Proc. of WACV*, pages 2339–2348, 2022. 4

[58] Wu Zheng, Li Jiang, Fanbin Lu, Yangyang Ye, and Chi-Wing Fu. Boosting Single-Frame 3D Object Detection by Simulating Multi-Frame Point Clouds. In *Proc. of ACM Multimedia*, pages 4848–4856, 2022. 3

[59] Brady Zhou and Philipp Krähenbühl. Cross-View Transformers for Real-Time Map-View Semantic Segmentation. In *Proc. of CVPR*, pages 13760–13769, 2022. 2

[60] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *Proc. of CVPR*, pages 4490–4499, 2018. 1

[61] Zheyuan Zhou, Liang Du, Xiaoqing Ye, Zhikang Zou, Xiao Tan, Li Zhang, Xiangyang Xue, and Jianfeng Feng. SGM3D: Stereo Guided Monocular 3D Object Detection. *IEEE Robotics and Automation Letters*, 7(4):10478–10485, 2022. 3

[62] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv preprint arXiv:1908.09492*, 2019. 6