BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation

Zhijian Liu[*]
MIT

Haotian Tang[*]
MIT

Alexander Amini
MIT

Xinyu Yang
Shanghai Jiao Tong University

Huizi Mao
OmniML

Daniela Rus
MIT

Song Han
MIT

https://bevfusion.mit.edu

Abstract

Multi-sensor fusion is essential for an accurate and reliable autonomous driving system. Recent approaches are based on point-level fusion: augmenting the Li-DAR point cloud with camera features. However, the camera-to-LiDAR projection throws away the semantic density of camera features, hindering the effectiveness of such methods, especially for semantic-oriented tasks (such as 3D scene segmentation). In this paper, we break this deeply-rooted convention with BEVFusion, an efficient and generic multi-task multi-sensor fusion framework. It unifies multi-modal features in the shared bird's-eye view (BEV) representation space, which nicely preserves both geometric and semantic information. To achieve this, we diagnose and lift key efficiency bottlenecks in the view transformation with optimized BEV pooling, reducing latency by more than $40\times$. BEVFusion is fundamentally task-agnostic and seamlessly supports different 3D perception tasks with almost no architectural changes. It establishes the new state of the art on nuScenes, achieving **1.3%** higher mAP and NDS on 3D object detection and **13.6%** higher mIoU on BEV map segmentation, with **1.9×** lower computation cost. Code to reproduce our results is available at https://github.com/mit-han-lab/bevfusion.

1 Introduction

Autonomous driving systems are equipped with diverse sensors. For instance, Waymo's self-driving vehicles have 29 cameras, 6 radars, and 5 LiDARs. Different sensors provide complementary signals: *e.g.*, cameras capture rich semantic information, LiDARs provide accurate spatial information, while radars offer instant velocity estimation. Therefore, multi-sensor fusion is of great importance for accurate and reliable perception.

Data from different sensors are expressed in fundamentally different modalities: *e.g.*, cameras capture data in perspective view and LiDAR in 3D view. To resolve this view discrepancy, we have to find a *unified representation* that is suitable for multi-task multi-modal feature fusion. Due to the tremendous success in 2D perception, the natural idea is to project the LiDAR point cloud onto the camera and process the RGB-D data with 2D CNNs. However, this LiDAR-to-camera projection introduces severe geometric distortion (see Figure 1a), which makes it less effective for geometric-oriented tasks, such as 3D object recognition.

Recent sensor fusion methods follow the other direction. They augment the LiDAR point cloud with semantic labels [54], CNN features [55, 23] or virtual points from 2D images [68], and then apply

---

[*] indicates equal contributions. The first two authors are listed in the alphabetical order.

(a) To **Camera**: *geometric*-lossy   (b) To **LiDAR**: *semantic*-lossy   (c) Shared   **BEVFusion**   space
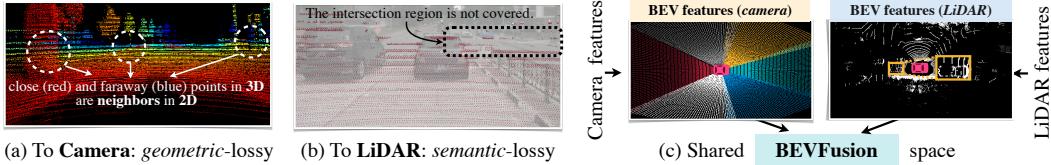
Figure 1: BEVFusion unifies camera and LiDAR features in a *shared* BEV space instead of mapping one modality to the other. It preserves camera's *semantic density* and LiDAR's *geometric structure*.

an existing LiDAR-based detector to predict 3D bounding boxes. Although they have demonstrated remarkable performance on large-scale detection benchmarks, these point-level fusion methods barely work on semantic-oriented tasks, such as BEV map segmentation [37, 39, 22, 70]. This is because the camera-to-LiDAR projection is semantically lossy (see Figure 1b): for a typical 32-beam LiDAR scanner, only 5% camera features will be matched to a LiDAR point while all others will be dropped. Such density differences will become even more drastic for sparser LiDARs (or imaging radars).

In this paper, we propose BEVFusion to unify multi-modal features in a shared bird's-eye view (BEV) representation space for task-agnostic learning. We maintain both geometric structure and semantic density (see Figure 1c) and naturally support most 3D perception tasks (since their output space can be naturally captured in BEV). While converting all features to BEV, we identify the major prohibitive efficiency bottleneck in the view transformation: *i.e.*, the BEV pooling operation alone takes more than 80% of the model's runtime. Then, we propose a specialized kernel with precomputation and interval reduction to eliminate this bottleneck, achieving more than **40×** speedup. Finally, we apply the fully-convolutional BEV encoder to fuse the unified BEV features and append a few task-specific heads to support different target tasks.

BEVFusion sets the new state-of-the-art performance on the nuScenes benchmark. On 3D object detection, it ranks **1st** on the leaderboard among all solutions. BEVFusion demonstrates even more significant improvements on BEV map segmentation. It achieves **6%** higher mIoU than camera-only models and **13.6%** higher mIoU than LiDAR-only models, while existing fusion methods hardly work. BEVFusion is efficient, delivering all these results with **1.9×** lower computation cost.

BEVFusion breaks the long-standing belief that point-level fusion is the best solution to multi-sensor fusion. Simplicity is also its key strength. We hope this work will serve as a simple yet strong baseline for future sensor fusion research and inspire the researchers to rethink the design and paradigm for generic multi-task multi-sensor fusion.

2 Related Work

LiDAR-Based 3D Perception. Researchers have designed single-stage 3D object detectors [72, 21, 65, 73, 66, 71] that extract flattened point cloud features using PointNets [41] or SparseConvNet [17] and perform detection in the BEV space. Later, Yin *et al.* [67] and others [15, 5, 42, 14, 6, 60] have explored anchor-free 3D object detection. Another stream of research [49, 10, 50, 47, 48, 24] focuses on two-stage object detection, which adds an RCNN network to existing one-stage object detectors. There are also U-Net like models specialized for 3D semantic segmentation [17, 13, 52, 33, 75], an important task for offline HD map construction.

Camera-Based 3D Perception. Due to the high cost of LiDAR sensors, researchers have spent significant efforts on camera-only 3D perception. FCOS3D [57] extends image detectors [53] with additional 3D regression branches, which is later improved in terms of depth modeling [58, 4]. Instead of performing object detection in the perspective view, DETR3D [59], PETR [30] and Graph-DETR3D [11] design DETR [74, 61]-based detection heads with learnable object queries in the 3D space. Inspired by the design of LiDAR-based detectors, another type of camera-only 3D perception models explicitly converts the camera features from perspective view to the bird's-eye view using a view transformer [37, 46, 45, 39]. BEVDet [20] and M^{2}BEV [63] effectively extend LSS [39] and OFT [46] to 3D object detection, achieving state-of-the-art performance upon release. CaDDN [43] adds explicit depth estimation supervision to the view transformer. BEVDet4D [19], BEVFormer [25] and PETRv2 [31] exploit temporal cues in multi-camera 3D object detection, achieving significant
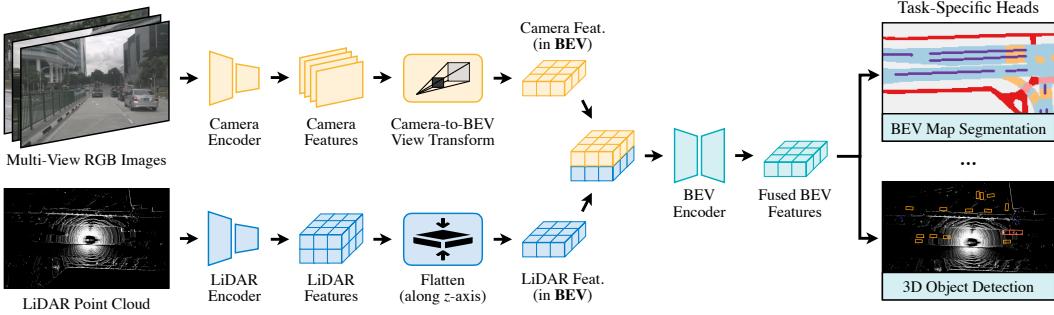
Figure 2: **BEVFusion** extracts features from multi-modal inputs and converts them into a shared bird's-eye view (BEV) space efficiently using view transformations. It fuses the unified BEV features with a fully-convolutional BEV encoder and supports different tasks with task-specific heads.

improvement over single-frame methods. BEVFormer [25], CVT [70] and Ego3RT [35] also study using multi-head attention to perform the view transformation.

Multi-Sensor Fusion. Recently, multi-sensor fusion arouses increased interest in the 3D detection community. Existing approaches can be classified into *proposal-level* and *point-level* fusion methods. MV3D [7] creates object proposals in 3D and projects the proposals to images to extract RoI features. F-PointNet [40], F-ConvNet [62] and CenterFusion [36] all lift image proposals into a 3D frustum. Lately, FUTR3D [8] and TransFusion [1] define object queries in the 3D space and fuse image features onto these proposals. Proposal-level fusion methods are *object-centric* and cannot trivially generalize to other tasks such as BEV map segmentation. Point-level fusion methods, on the other hand, usually paint image semantic features onto foreground LiDAR points and perform LiDAR-based detection on the decorated point cloud inputs. As such, they are both *object-centric* and *geometric-centric*. Among all these methods, PointPainting [54], PointAugmenting [55], MVP [68], FusionPainting [64], AutoAlign [12] and FocalSparseCNN [9] are (LiDAR) input-level decoration, while Deep Continuous Fusion [27] and DeepFusion [23] are feature-level decoration.

Multi-Task Learning. Multi-task learning have been well-studied in the computer vision community. Researchers have studied to jointly perform object detection and instance segmentation [44, 3] and have extended to pose estimation and human-object interaction [18, 51, 56, 16]. A few concurrent works including M^{2}BEV [63], BEVFormer [25] and BEVerse [69] jointly perform object detection and BEV segmentation in 3D. None of the above methods considers multi-sensor fusion. MMF [26] simultaneously works on depth completion and object detection with both camera and LiDAR inputs, but is still object-centric and not applicable to BEV map segmentation.

In contrast to all existing methods, BEVFusion performs sensor fusion in a shared BEV space and treats foreground and background, geometric and semantic information equally. BEVFusion is a generic multi-task multi-sensor perception framework.

3 Method

BEVFusion focuses on *multi-sensor fusion* (*i.e.*, multi-view cameras and LiDAR) for *multi-task 3D perception* (*i.e.*, detection and segmentation). We provide an overview of our framework in Figure 2. Given different sensory inputs, we first apply modality-specific encoders to extract their features. We transform multi-modal features into a unified BEV representation that preserves both geometric and semantic information. We identify the efficiency bottleneck of the view transformation and accelerate BEV pooling with precomputation and interval reduction. We then apply the convolution-based BEV encoder to the unified BEV features to alleviate the local misalignment between different features. Finally, we append a few task-specific heads to support different 3D tasks.

3.1 Unified Representation

Different features can exist in different views. For instance, camera features are in the perspective view, while LiDAR/radar features are typically in the 3D/bird's-eye view. Even for camera features, each one of them has a distinct viewing angle (*i.e.*, front, back, left, right). This *view discrepancy*
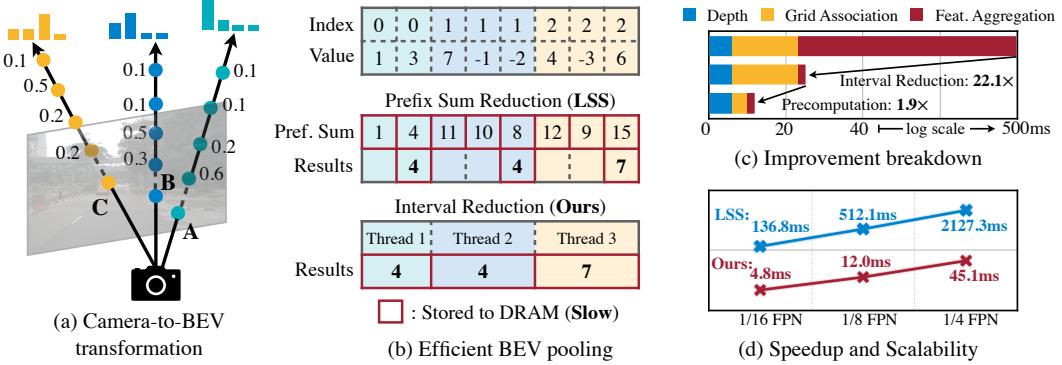
Figure 3: Camera-to-BEV transformation (a) is the key step to perform sensor fusion in the unified BEV space. However, existing implementation is extremely slow and can take up to 2s for a single scene. We propose efficient BEV pooling (b) using interval reduction and fast grid association with precomputation, bringing about **40×** speedup to the view transformation module (c, d).

makes the feature fusion difficult since the same element in different feature tensors might correspond to completely different spatial locations (and the naïve elementwise feature fusion will not work in this case). Therefore, it is crucial to find a *shared* representation, such that (1) all sensor features can be easily converted to it without information loss, and (2) it is suitable for different types of tasks.

To Camera. Motivated by RGB-D data, one choice is to project the LiDAR point cloud to the camera plane and render the 2.5D sparse depth. However, this conversion is *geometrically lossy*. Two neighbors on the depth map can be far away from each other in the 3D space. This makes the camera view less effective for tasks that focus on the object/scene geometry, such as 3D object detection.

To LiDAR. Most state-of-the-art sensor fusion methods [54, 68, 23] decorate LiDAR points with their corresponding camera features (*e.g.*, semantic labels, CNN features or virtual points). However, this camera-to-LiDAR projection is *semantically lossy*. Camera and LiDAR features have drastically different densities, resulting in only less than 5% of camera features being matched to a LiDAR point (for a 32-channel LiDAR scanner). Giving up the semantic density of camera features severely hurts the model's performance on semantic-oriented tasks (such as BEV map segmentation). Similar drawbacks also apply to more recent fusion methods in the latent space (*e.g.*, object query) [8, 1].

To Bird's-Eye View. We *adopt the bird's-eye view (BEV) as the unified representation for fusion*. This view is friendly to almost all perception tasks since the output space is also in BEV. More importantly, the transformation to BEV keeps both geometric structure (from LiDAR features) and semantic density (from camera features). On the one hand, the LiDAR-to-BEV projection flattens the sparse LiDAR features along the height dimension, thus does not create geometric distortion in Figure 1a. On the other hand, camera-to-BEV projection casts each camera feature pixel back into a ray in the 3D space (detailed in the next section), which can result in a dense BEV feature map in Figure 1c that retains full semantic information from the cameras.

3.2 Efficient Camera-to-BEV Transformation

Camera-to-BEV transformation is non-trivial because the depth associated with each camera feature pixel is inherently ambiguous. Following LSS [39] and BEVDet [20, 19], we explicitly predict the discrete depth distribution of each pixel. We then scatter each feature pixel into D discrete points along the camera ray and rescale the associated features by their corresponding depth probabilities (Figure 3a). This generates a camera feature point cloud of size $NHWD$, where N is the number of cameras and (H, W) is the camera feature map size. Such 3D feature point cloud is quantized along the x, y axes with a step size of r (*e.g.*, 0.4m). We use the *BEV pooling* operation to aggregate all features within each $r \times r$ BEV grid and flatten the features along the z-axis.

Though simple, BEV pooling is surprisingly inefficient and slow, taking more than 500ms on an RTX 3090 GPU (while the rest of our model only takes around 100ms). This is because the camera feature

point cloud is very large: for a typical workload[*], there could be around 2 million points generated for each frame, two orders of magnitudes denser than a LiDAR feature point cloud. To lift this efficiency bottleneck, we propose to optimize the BEV pooling with precomputation and interval reduction.

Precomputation. The first step of BEV pooling is to *associate* each point in the camera feature point cloud with a BEV grid. Different from LiDAR point clouds, the coordinates of the camera feature point cloud are *fixed* (as long as the camera intrinsics and extrinsics stay the same, which is usually the case after proper calibration). Motivated by this, we precompute the 3D coordinate and the BEV grid index of each point. We also sort all points according to grid indices and record the rank of each point. During inference, we only need to reorder all feature points based on the precomputed ranks. This caching mechanism can reduce the latency of grid association from 17ms to 4ms.

Interval Reduction. After grid association, all points within the same BEV grid will be consecutive in the tensor representation. The next step of BEV pooling is then to *aggregate* the features within each BEV grid by some symmetric function (*e.g.*, mean, max, and sum). As in Figure 3b, existing implementation [39] first computes the prefix sum over all points and then subtracts the values at the boundaries where indices change. However, the prefix sum operation requires tree reduction on the GPU and produces many unused partial sums (since we only need those values on the boundaries), both of which are inefficient. To accelerate feature aggregation, we implement a specialized GPU kernel that parallelizes directly over BEV grids: we assign a GPU thread to each grid that calculates its interval sum and writes the result back. This kernel removes the dependency between outputs (thus does not require multi-level tree reduction) and avoids writing the partial sums to the DRAM, reducing the latency of feature aggregation from 500ms to 2ms (Figure 3c).

Takeaways. The camera-to-BEV transformation is **40×** faster with our optimized BEV pooling: the latency is reduced from more than 500ms to 12ms (only 10% of our model's end-to-end runtime) and scales well across different feature resolutions (Figure 3d). This is a key enabler for unifying multi-modal sensory features in the shared BEV representation. Two concurrent works of ours also identify this efficiency bottleneck in the camera-only 3D detection. They approximate the view transformer by assuming uniform depth distribution [63] or truncating the points within each BEV grid [20]. In contrast, our techniques are *exact* without any approximation, while still being faster.

3.3 Fully-Convolutional Fusion

With all sensory features converted to the shared BEV representation, we can easily fuse them together with an elementwise operator (such as concatenation). Though in the same space, LiDAR BEV features and camera BEV features can still be spatially misaligned to some extent due to the inaccurate depth in the view transformer. To this end, we apply a convolution-based BEV encoder (with a few residual blocks) to compensate for such local misalignments. Our method could potentially benefit from more accurate depth estimation (*e.g.*, supervising the view transformer with ground-truth depth [43, 38]), which we leave for future work.

3.4 Multi-Task Heads

We apply multiple task-specific heads to the fused BEV feature map. Our method is applicable to most 3D perception tasks. We showcase two examples: 3D object detection and BEV map segmentation.

Detection. We use a class-specific center heatmap head to predict the center location of all objects and a few regression heads to estimate the object size, rotation, and velocity. We refer the readers to previous 3D detection papers [1, 67, 68] for more details.

Segmentation. Different map categories may overlap (*e.g.*, crosswalk is a subset of drivable space). Therefore, we formulate this problem as multiple binary semantic segmentation, one for each class. We follow CVT [70] to train the segmentation head with the standard focal loss [29].

4 Experiments

We evaluate BEVFusion for camera-LiDAR fusion on 3D object detection and BEV map segmentation, covering both geometric- and semantic-oriented tasks. Our framework can be easily extended

---

[*]$N = 6$, $(H, W) = (32, 88)$, and $D = (60 - 1)/0.5 = 118$. This corresponds to six multi-view cameras, each associated with a 32×88 camera feature map (which is downsampled from a 256×704 image by $8 \times$). The depth is discretized into $[1, 60]$ meters with a step size of 0.5 meter, following BEVDet [20].

Table 1: BEVFusion achieves state-of-the-art 3D object detection performance on nuScenes (`val` and `test`) without bells and whistles. It breaks the convention of decorating camera features onto the LiDAR point cloud and delivers at least 1.3% higher mAP and NDS with **1.5-2×** lower computation cost. (*: our re-implementation; †: w/ test-time augmentation (TTA); ‡: w/ model ensemble and TTA)

	Modality	mAP (*test*)	NDS (*test*)	mAP (*val*)	NDS (*val*)	MACs (G)	Latency (ms)
BEVDet [20]	C	42.2†	48.2†	–	–	–	–
M²BEV [63]	C	42.9	47.4	41.7	47.0	–	–
BEVFormer [25]	C	44.5	53.5	41.6	51.7	–	–
BEVDet4D [19]	C	45.1†	56.9†	–	–	–	–
PointPillars [21]	L	–	–	52.3	61.3	65.5	34.4
SECOND [65]	L	52.8	63.3	52.6	63.0	85.0	69.8
CenterPoint [67]	L	60.3	67.3	59.6	66.8	153.5	80.7
PointPainting [54]	C+L	–	–	65.8*	69.6*	370.0	185.8
PointAugmenting [55]	C+L	66.8†	71.0†	–	–	408.5	234.4
MVP [68]	C+L	66.4	70.5	66.1*	70.0*	371.7	187.1
FusionPainting [64]	C+L	68.1	71.6	66.5	70.7	–	–
AutoAlign [12]	C+L	–	–	66.6	71.1	–	–
FUTR3D [8]	C+L	–	–	64.5	68.3	1069.0	321.4
TransFusion [1]	C+L	68.9	71.6	67.5	71.3	485.8	156.6
BEVFusion (Ours)	C+L	**70.2**	**72.9**	**68.5**	**71.4**	**253.2**	**119.2**
CenterPoint-Fusion	C+R+L	72.4‡	74.9‡	–	–	–	–
FusionVPE	C+L	73.3‡	75.5‡	–	–	–	–
BEVFusion (Ours)	C+L	**75.0‡**	**76.1‡**	**73.7‡**	**74.9‡**	–	–

Table 2: BEVFusion outperforms the state-of-the-art multi-sensor fusion methods by **13.6%** on BEV map segmentation on nuScenes (`val`) with consistent improvements across different categories.

	Modality	Drivable	Ped. Cross.	Walkway	Stop Line	Carpark	Divider	Mean
OFT [46]	C	74.0	35.3	45.9	27.5	35.9	33.9	42.1
LSS [39]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT [70]	C	74.3	36.8	39.9	25.8	35.0	29.4	40.2
M²BEV [63]	C	77.2	–	–	–	–	40.5	–
BEVFusion (Ours)	C	**81.7**	**54.8**	**58.4**	**47.4**	**50.7**	**46.4**	**56.6**
PointPillars [21]	L	72.0	43.1	53.1	29.7	27.7	37.5	43.8
CenterPoint [67]	L	75.6	48.4	57.5	36.5	31.7	41.9	48.6
PointPainting [54]	C+L	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP [68]	C+L	76.1	48.7	57.0	36.9	33.0	42.2	49.0
BEVFusion (Ours)	C+L	**85.5**	**60.5**	**67.6**	**52.0**	**57.0**	**53.7**	**62.7**

to support other types of sensors (such as radars and event-based cameras) and other 3D perception tasks (such as 3D object tracking and motion forecasting).

Model. We use Swin-T [32] as our image backbone and VoxelNet [65] as our LiDAR backbone. We apply FPN [28] to fuse multi-scale camera features to produce a feature map of 1/8 input size. We downsample camera images to 256×704 and voxelize the LiDAR point cloud with 0.075m (for detection) and 0.1m (for segmentation). As detection and segmentation tasks require BEV feature maps with different spatial ranges and sizes, we apply grid sampling with bilinear interpolation before each task-specific head to explicitly transform between different BEV feature maps.

Training. Unlike existing approaches [54, 55, 1] that freeze the camera encoder, we train the entire model in an end-to-end manner. We apply both image and LiDAR data augmentations to prevent overfitting. Optimization is carried out using AdamW [34] with a weight decay of 10^{-2}.

Dataset. We evaluate our method on nuScenes [2] which is a large-scale outdoor dataset released under the CC BY-NC-SA 4.0 license. It has diverse annotations to support all sorts of tasks (such as 3D object detection/tracking and BEV map segmentation). Each of the 40,157 annotated samples contains six monocular camera images with 360-degree FoV and a 32-beam LiDAR scan.
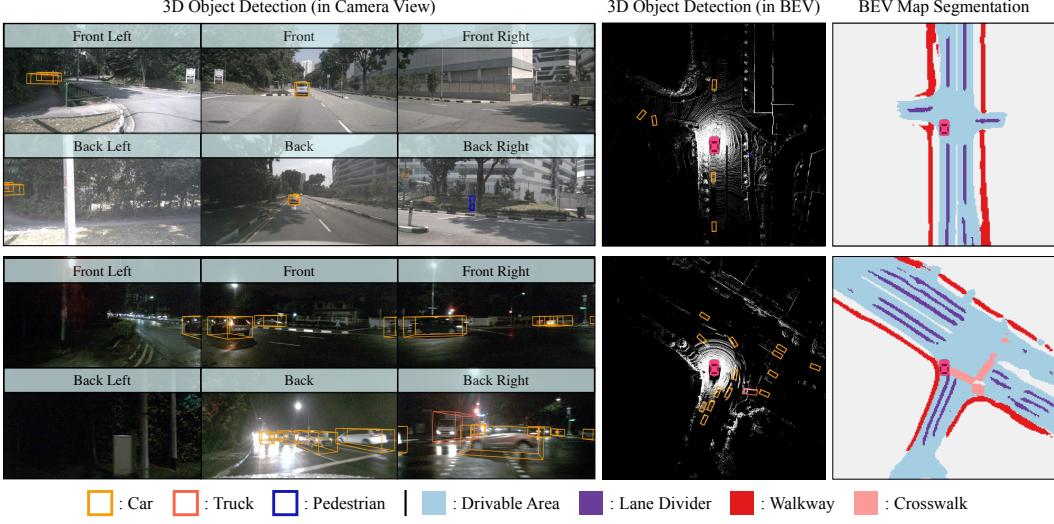
Figure 4: Qualitative results of BEVFusion on 3D object detection and BEV map segmentation. It accurately recognizes distant and small objects (top) and parses crowded nighttime scenes (bottom).

4.1 3D Object Detection

We first experiment on the geometric-centric 3D object detection benchmark, where BEVFusion achieves superior performance with lower computation cost and measured latency.

Setting. We use the mean average precision (mAP) across 10 foreground classes and the nuScenes detection score (NDS) as our detection metrics. We also measure the single-inference #MACs and latency on an RTX3090 GPU for all open-source methods. We use a single model without any test-time augmentation for both `val` and `test` results.

Results. As in Table 1, BEVFusion achieves state-of-the-art results on the nuScenes detection benchmark, with close-to-real-time (**8.4 FPS**) inference speed on a desktop GPU. Compared with TransFusion [1], BEVFusion achieve 1.3% improvement in `test` split mAP and NDS, while significantly reduces the MACs by **1.9×** and measured latency by **1.3×**. BEVFusion also compares favorably against representative point-level fusion methods PointPainting [54] and MVP [68] with **1.6×** speedup, **1.5×** MACs reduction and **3.8%** higher mAP on the `test` set. We argue that the efficiency gain of BEVFusion comes from the fact that we choose the BEV space as the share fusion space, which fully utilizes all camera features instead of just a 5% sparse set. As a result, BEVFusion can achieve the same performance with much smaller MACs. Combined with the efficient BEV pooling operator in Section 3.2, BEVFusion translates MACs reduction into measured speedup.

4.2 BEV Map Segmentation

We further compare BEVFusion with state-of-the-art 3D perception models on the semantic-centric BEV map segmentation task, where BEVFusion achieves an even larger performance boost.

Setting. We report the Intersection-over-Union (IoU) on 6 background classes (drivable space, pedestrian crossing, walkway, stop line, car-parking area, and lane divider) and the class-averaged mean IoU as our evaluation metric. As different classes may have overlappings (*e.g.* car-parking area is also drivable), we evaluate the binary segmentation performance for each class separately and select the highest IoU across different thresholds [70]. For each frame, we only perform the evaluation in the [-50m, 50m]×[-50m, 50m] region around the ego car following [39, 70, 63, 25]. In BEVFusion, we use a single model that jointly performs binary segmentation for all classes instead of following the conventional approach to train a separate model for each class. This results in **6×** faster inference and training. We reproduced the results of all open-source competing methods.

Results. We report the BEV map segmentation results in Table 2. In contrast to 3D object detection which is a *geometric*-oriented task, map segmentation is *semantic*-oriented. As a result, our camera-only BEVFusion model outperforms LiDAR-only baselines by **8-13%**. This observation is the exact opposite of results in Table 1, where state-of-the-art camera-only 3D detectors got outperformed by LiDAR-only detectors by almost 20 mAP. Our camera-only model boosts the performance of

Table 3: BEVFusion is robust under different lighting and weather conditions, significantly boosting the performance of single-modality baselines (marked in gray) under challenging rainy and nighttime scenarios. (*: variants of BEVDet-Tiny and LSS with larger backbones and view transformers)

	Modality	Sunny		Rainy		Day		Night	
		mAP	mIoU	mAP	mIoU	mAP	mIoU	mAP	mIoU
CenterPoint	L	62.9	50.7	59.2	42.3	62.8	48.9	35.4	37.0
BEVDet/LSS*	C	32.9	59.0	33.7	50.5	33.7	57.4	13.5	30.8
MVP	C+L	65.9 (+3.0)	51.0 (-8.0)	66.3 (+7.1)	42.9 (-7.6)	66.3 (+3.5)	49.2 (-8.2)	38.4 (+3.0)	37.5 (+6.7)
BEVFusion	C+L	68.2 (+5.3)	65.6 (+6.6)	69.9 (+10.7)	55.9 (+5.4)	68.5 (+5.7)	63.1 (+5.7)	42.8 (+7.4)	43.6 (+12.8)

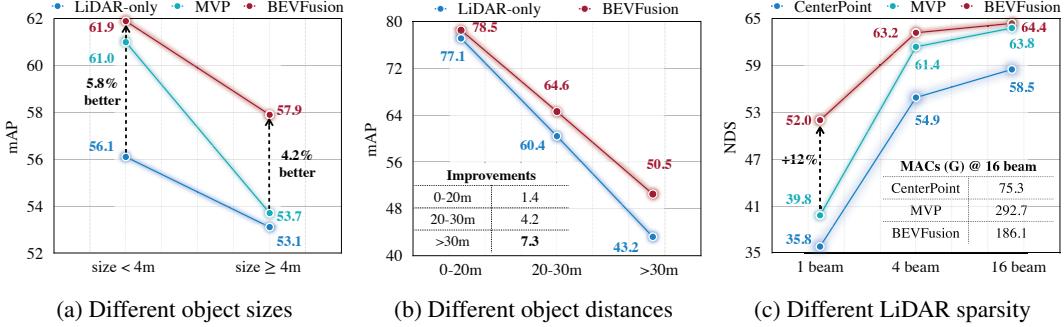(a) Different object sizes (b) Different object distances (c) Different LiDAR sparsity

Figure 5: BEVFusion consistently outperforms state-of-the-art single- and multi-modality detectors under different LiDAR sparsity, object sizes and object distances from the ego car, especially under the more challenging settings (*i.e.*, *sparser point clouds*, *small/distant objects*).

existing monocular BEV map segmentation methods by at least **12%**. In the multi-modality setting, we further improve the performance of the monocular BEVFusion by **6** mIoU and achieved >**13%** improvement over state-of-the-art sensor fusion methods [54, 68]. This is because both baseline methods are *object*-centric and *geometric*-oriented. PointPainting [54] only decorates the *foreground* LiDAR points and MVP only densifies *foreground* 3D objects. Both approaches are not helpful for segmenting map components. Worse still, both methods assume that LiDAR should be the more effective modality in sensor fusion, which is not true according to our observations in Table 2.

5 Analysis

We present in-depth analyses of BEVFusion over single-modality models and state-of-the-art multi-modality models under different circumstances.

Weather and Lighting. We systematically analyze the performance of BEVFusion under different weather and lighting conditions in Table 3. Detecting objects in rainy weather is challenging for LiDAR-only models due to significant sensor noises. Thanks to the robustness of camera sensors under different weathers, BEVFusion improves CenterPoint by **10.7** mAP, closing the performance gap between sunny and rainy scenarios. Poor lighting conditions are challenging for both detection and segmentation models. For detection, MVP achieves a much smaller improvement compared to BEVFusion since it requires *accurate* 2D instance segmentations to generate multi-modal virtual points (MVPs). This can be very challenging in dark or overexposed scenes (*e.g.*, the second scene of Figure 4). For segmentation, even if the camera-only BEVFusion greatly outperforms CenterPoint on the entire dataset in Table 2, its performance is much worse at nighttime. Our BEVFusion significantly boosts its performance by **12.8** mIoU, which is even larger than the improvement in the daytime, demonstrating the significance of geometric clues when camera sensors fail.

Sizes and Distances. We also analyze the performance under different object sizes and distances. From Figure 5a, BEVFusion achieves consistent improvements over its LiDAR-only counterpart for both small and large objects, while MVP has only negligible improvements for objects larger than 4m. This is because larger objects are typically much denser, benefiting less from those augmented multi-modal virtual points (MVPs). Besides, BEVFusion brings larger improvements to the LiDAR-

8

Table 4: Ablation experiments to validate our design choices. Default settings are marked in `gray`.

(a) Modality	mAP	NDS	mIoU
L	57.6	64.9	48.6
C	33.3	40.2	56.6
L+C	**66.4**	**69.5**	**62.7**

(b) Voxel size	mAP	NDS	mIoU
0.075	**67.1**	**70.2**	60.6
0.1	66.4	69.5	62.7
0.125	65.1	68.6	**63.7**

(c) Image size	mAP	NDS	mIoU
128×352	64.0	68.2	60.5
256×704	66.4	69.5	**62.7**
384×1056	**67.2**	**70.0**	59.9

(d) FPN size	mAP	NDS	mIoU
1/16	66.2	69.4	**62.7**
1/8	**66.4**	**69.5**	**62.7**
1/4	66.0	69.2	58.8

(e) Data augmentation	mAP	NDS	mIoU
Image	63.8	68.1	62.3
LiDAR	65.7	69.1	61.3
Both	**66.4**	**69.5**	**62.7**

(f) Image backbone	mAP	NDS	mIoU
ResNet50	65.3	68.9	59.2
SwinT (freeze)	66.1	69.3	52.8
SwinT	**66.4**	**69.5**	**62.7**

only model for smaller objects (Figure 5a) and more distant objects (Figure 5b), both of which are poorly covered by LiDAR and can therefore benefit more from the dense camera information.

Sparser LiDARs. We demonstrate the performance of the LiDAR-only detector CenterPoint [67], multi-modality detector MVP [68] and our BEVFusion under different LiDAR sparsity in Figure 5c. BEVFusion consistently outperforms MVP under all sparsity levels with **1.6×** MACs reduction and achieves a **12%** improvement in the 1-beam LiDAR scenario. MVP decorates the input point cloud and directly applies CenterPoint on the painted and densified LiDAR input. Thus, it naturally requires the LiDAR-only CenterPoint detector to perform well, which is not valid under sparse LiDAR settings (35.8 NDS with 1-beam input in Figure 5c). BEVFusion, in contrast, fuses multi-sensory information in a shared BEV space, and thus does not assume a strong LiDAR-only detector.

Multi-Task Learning. This paper focuses on the setting where different tasks are trained separately. Here, we present a pilot study of joint 3D detection and segmentation training. We re-scale the loss for different tasks to the same magnitude and apply a separate BEV encoder for each task to provide the capability of learning more task-specific features. From Table 5, jointly training different tasks together has a negative impact on the performance of each individual task, which is widely known as "negative transfer". Separating BEV encoders partially alleviates this problem. A more sophisticated training scheme could further close this gap, which we leave for future work.

Table 5: Joint detection and segmentation training (trained for 10 epochs).

	NDS	mIoU
Detection only	70.4	–
Segmentation only	–	58.5
Joint (*shared BEV encoders*)	69.7	54.0
Joint (*separate BEV encoders*)	**69.9**	**58.4**

Ablation Studies. We present ablation studies in Table 4 to justify our design choices, where we use a shorter training schedule for detectors. In Table 4a, we observe that BEVFusion brings large improvements to both LiDAR-only detection (**+8.8%**) and camera-only segmentation (**+6.1%**). This indicates that sensor fusion in a shared BEV space is beneficial for both geometric and semantic-oriented tasks. Table 4b, Table 4c and Table 4d suggest that the detection variant of BEVFusion scales well for both voxel and image resolutions, while the BEV segmentation performance plateaus when the image resolution grows above 256×704. We also notice in Table 4d that using FPN features from 1/8 input resolution provides the best performance for both detection and segmentation and further increasing computation is not helpful. Table 4f indicates that our BEVFusion is general and works well for different backbones. It is also noteworthy that the common practice to freeze the image backbone in existing multi-sensor 3D object detection research [54, 55, 1] does not exploit the full potential of the camera feature extractor even for detection, and causes a drastic performance drop (**10%**) in BEV segmentation. We further demonstrate in Table 4e that augmentations on both image and LiDAR inputs are helpful for improving the performance on BEVFusion.

6 Conclusion

We present BEVFusion, an efficient and generic framework for multi-task multi-sensor 3D perception. BEVFusion unifies camera and LiDAR features in a shared BEV space that fully preserves both geometric and semantic information. To achieve this, we accelerate the slow camera-to-BEV transformation by more than 40×. BEVFusion breaks the long-lasting common practice that point-level fusion is the golden choice for multi-sensor perception systems. BEVFusion achieves state-of-the-art performance on both 3D detection and BEV map segmentation tasks with 1.5-1.9× less computation

and 1.3-1.6× measured speedup over existing solutions. We hope that BEVFusion can serve as a simple but powerful baseline to inspire future research on multi-task multi-sensor fusion.

Limitations. At present, BEVFusion still has performance degradation in joint multi-task training, which has not yet unlocked the potential for larger inference speedup in the multi-task setting. More accurate depth estimation [43, 38] is also an under-explored direction in this paper that can potentially boost the performance of BEVFusion further.

Societal Impacts. Efficient and accurate multi-sensor perception is crucial for the safety of autonomous vehicles. BEVFusion reduces the computation cost of state-of-the-art multi-sensor fusion models by half and achieves large accuracy improvements on small and distant objects, and in rainy and night conditions. It paves the way for safe and robust autonomous driving.

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 *CVPR*, 2022. 3, 4, 5, 6, 7, 9

[2] 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 *CVPR*, 2020. 6

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. In *CVPR*, 2018. 3

[4] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation. In *CVPR*, 2022. 2

[5] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as Hotspots: An Anchor-Free 3D Object Detection Approach via Firing of Hotspots. In *ECCV*, 2020. 2

[6] Qi Chen, Sourabh Vora, and Oscar Beijbom. PolarStream: Streaming Lidar Object Detection and Segmentation with Polar Pillars. In *NeurIPS*, 2021. 2

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-View 3D Object Detection Network for Autonomous Driving. In *CVPR*, 2017. 3

[8] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. FUTR3D: A Unified Sensor Fusion Framework for 3D Detection. *arXiv*, 2022. 3, 4, 6

[9] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal Sparse Convolutional Networks for 3D Object Detection. In *CVPR*, 2022. 3

[10] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast Point R-CNN. In *ICCV*, 2019. 2

[11] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinghong Jiang, and Feng Zhao. Graph-DETR3D: Rethinking Overlapping Regions for Multi-View 3D Object Detection. In *ACM-MM*, 2022. 2

[12] 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. *arXiv*, 2022. 3, 6

[13] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 2

[14] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection. In *ICCV*, 2021. 2

[15] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. 1st Place Solutions to the Real-time 3D Detection and the Most Efficient Model of the Waymo Open Dataset Challenge 2021. In *CVPRW*, 2021. 2

[16] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and Recognizing Human-Object Interactions. In *CVPR*, 2018. 3

[17] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *CVPR*, 2018. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 3

[19] Junjie Huang and Guan Huang. BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection. *arXiv*, 2022. 2, 4, 6

[20] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View. *arXiv*, 2021. 2, 4, 5, 6

[21] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, and Jiong Yang. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *CVPR*, 2019. 2, 6

[22] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An Online HD Map Construction and Evaluation Framework. In *ICRA*, 2022. 2

[23] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Bo Wu, Yifeng Lu, Denny Zhou, et al. DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection. In *CVPR*, 2022. 1, 3, 4

[24] Zhichao Li, Feng Wang, and Naiyan Wang. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. *CVPR*, 2021. 2

[25] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. *arXiv*, 2022. 2, 3, 6, 7

[26] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-Task Multi-Sensor Fusion for 3D Object Detection. In *CVPR*, 2019. 3

[27] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. In *ECCV*, 2018. 3

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 6

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 5

[30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position Embedding Transformation for Multi-View 3D Object Detection. *arXiv*, 2022. 2

[31] 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*, 2022. 2

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 2021. 6

[33] Zhijian Liu, Haotian Tang, Shengyu Zhao, Kevin Shao, and Song Han. PVNAS: 3D Neural Architecture Search with Point-Voxel Convolution. *TPAMI*, 2021. 2

[34] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 6

[35] Jiachen Lu, Zheyuan Zhou, Xiatian Zhu, Hang Xu, and Li Zhang. Learning Ego 3D Representation as Ray Tracing. *arXiv*, 2022. 3

[36] Ramin Nabati and Hairong Qi. CenterFusion: Center-Based Radar and Camera Fusion for 3D Object Detection. In *WACV*, 2021. 3

[37] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-View Semantic Segmentation for Sensing Surroundings. *RA-L*, 2020. 2

[38] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is Pseudo-Lidar needed for Monocular 3D Object detection? In *ICCV*, 2021. 5, 10

[39] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *ECCV*, 2020. 2, 4, 5, 6, 7

[40] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *CVPR*, 2018. 3

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 2

[42] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3D Object Detection from Point Cloud Sequences. In *CVPR*, 2021. 2

[43] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distributionnetwork for monocular 3d object detection. In *CVPR*, 2021. 2, 5, 10

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 3

[45] Thomas Roddick and Roberto Cipolla. Predicting Semantic Map Representations from Images using Pyramid Occupancy Networks. In *CVPR*, 2020. 2

[46] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic Feature Transform for Monocular 3D Object Detection. In *BMVC*, 2019. 2, 6

[47] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *CVPR*, 2020. 2

[48] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jinaping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *arXiv*, 2021. 2

[49] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *CVPR*, 2019. 2

[50] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *TPAMI*, 2020. 2

[51] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*, 2019. 3

[52] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *ECCV*, 2020. 2

[53] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*, 2019. 2

[54] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential Fusion for 3D Object Detection. In *CVPR*, 2020. 1, 3, 4, 6, 7, 8, 9

[55] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. PointAugmenting: Cross-Modal Augmentation for 3D Object Detection. In *CVPR*, 2021. 1, 3, 6, 9

[56] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019. 3

[57] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. In *ICCVW*, 2021. 2

[58] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting Objects in Perspective. In *CoRL*, 2021. 2

[59] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin M. Solomon. DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. In *CoRL*, 2021. 2

[60] Yue Wang and Justin M. Solomon. Object DGCNN: 3D Object Detection using Dynamic Graphs. In *NeurIPS*, 2021. 2

[61] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query Design for Transformer-Based Detector. In *AAAI*, 2022. 2

[62] Zhixin Wang and Kui Jia. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. In *IROS*, 2019. 3

[63] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M^{2}BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. *arXiv*, 2022. 2, 3, 5, 6, 7

[64] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Bin Zhou, and Liangjun Zhang. FusionPainting: Multimodal Fusion with Adaptive Attention for 3D Object Detection. In *ITSC*, 2021. 3, 6

[65] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 2018. 2, 6

[66] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-Based 3D Single Stage Object Detector. *CVPR*, 2020. 2

[67] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Based 3D Object Detection and Tracking. In *CVPR*, 2021. 2, 5, 6, 9

[68] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal Virtual Point 3D Detection. In *NeurIPS*, 2021. 1, 3, 4, 5, 6, 7, 8, 9

[69] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. BEVerse: Unified Perception and Prediction in Birds-Eye-View for Vision-Centric Autonomous Driving. *arXiv*, 2022. 3

[70] Brady Zhou and Philipp Krähenbühl. Cross-View Transformers for Real-Time Map-View Semantic Segmentation. In *CVPR*, 2022. 2, 3, 5, 6, 7

[71] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds. *CoRL*, 2019. 2

[72] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*, 2018. 2

[73] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-Balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv*, 2019. 2

[74] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021. 2

[75] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *CVPR*, 2021. 2