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

Zhijian Liu*,[1], Haotian Tang*,[1], Alexander Amini[1], Xinyu Yang[1], Huizi Mao[2], Daniela L. Rus[1], Song Han[1]

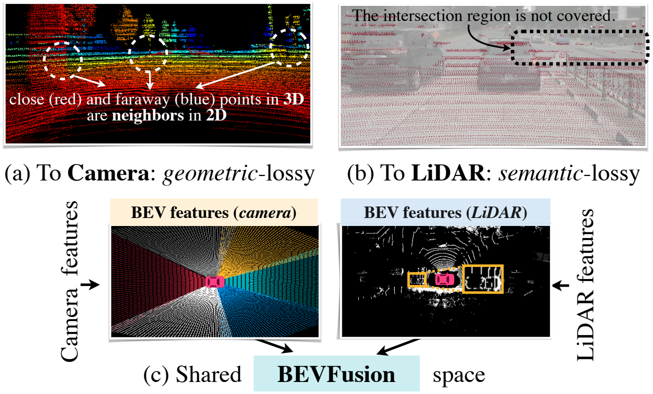
***Abstract*— Multi-sensor fusion is essential for an accurate and reliable autonomous driving system. Recent approaches are based on point-level fusion: augmenting the LiDAR point cloud with camera features. However, the camera-to-LiDAR projection throws away the semantic density of camera features, hindering the effectiveness of such methods, especially for semantic-oriented tasks (such as 3D scene segmentation). In this paper, we propose 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 the key efficiency bottlenecks in the view transformation with optimized BEV pooling, reducing latency by more than 40×. 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 the nuScenes benchmark, 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.**

I. 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. Thus, multi-sensor fusion is essential 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 [1], CNN features [2], [3] or virtual points from 2D images [4], and then apply 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 [5], [6], [7], [8]. 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 radars).


close (red) and faraway (blue) points in 3D are neighbors in 2D
The intersection region is not covered.
BEV features (camera)
BEV features (LiDAR)
Camera features
LiDAR features

(a) To **Camera**: *geometric*-lossy (b) To **LiDAR**: *semantic*-lossy

(c) Shared **BEVFusion** space

Fig. 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*.

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 3D object detection performance on both nuScenes and Waymo benchmarks. It outperforms all published methods with or without test-time augmentation and model ensemble. 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. Moreover, BEVFusion

* The first two authors contributed equally and are listed alphabetically. This work was supported by MIT-IBM Watson AI Lab, National Science Foundation, Hyundai Motor, Qualcomm, NVIDIA and Apple. Zhijian Liu was partially supported by the Qualcomm Innovation Fellowship.

[1] Z. Liu, H. Tang, A. Amini, X. Yang, D. Rus, and S. Han are with Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

[2] H. Mao is with OmniML, San Jose, CA 95131, USA.

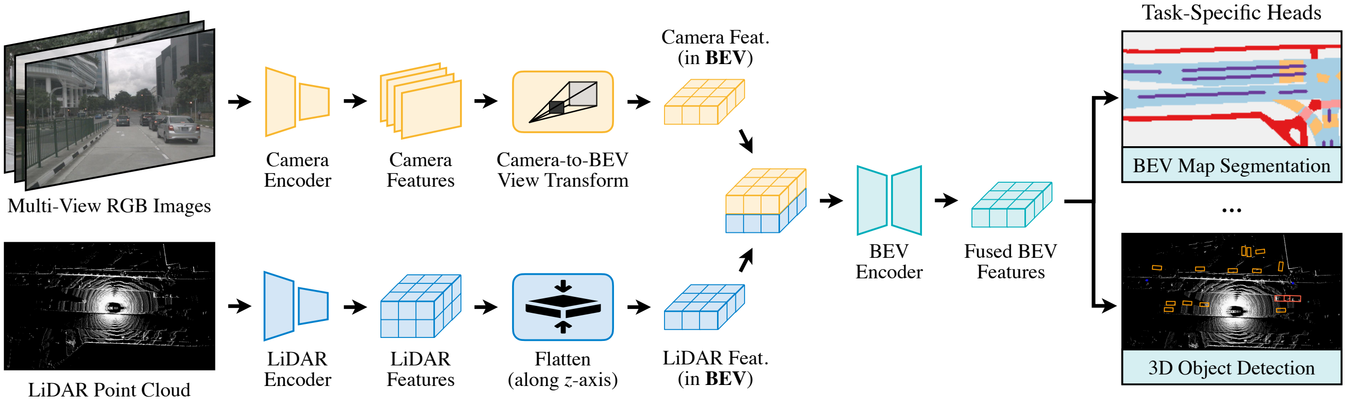


Fig. 2: **BEVFusion** extracts features from multi-modal inputs and converts them into a shared bird's-eye view (BEV) space efficiently using view transformations. It fuses the unified BEV features with a fully-convolutional BEV encoder and supports different tasks with task-specific heads.

is highly efficient, delivering all these results with **1.9**× lower computation cost.

While point-level fusion has been the go-to choice over the past three years, BEVFusion provides a fresh perspective to the field of multi-sensor fusion by rethinking "*Is LiDAR space the right place to perform sensor fusion?*". It showcases the superior performance of an alternative paradigm that has been previously overlooked. 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.

II. Related Work

LiDAR-Based 3D Perception. Researchers have designed single-stage 3D object detectors [9], [10], [11], [12], [13], [14] that extract flattened point cloud features using PointNets [15] or SparseConvNet [16] and perform detection in the BEV space. Later, [17], [18], [19], [20], [21], [22], [23] explore anchor-free single-stage 3D object detection. Another stream of research [24], [25], [26], [27], [28], [29] focuses on two-stage object detector design, which adds an RCNN network to existing one-stage object detectors.

Camera-Based 3D Perception. Due to the high cost of LiDAR sensors, researchers spend significant efforts on camera-only 3D perception. FCOS3D [30] extends an image detector [31] with additional 3D regression branches, which is later improved by [32], [33] in depth modeling. Instead of performing object detection in the perspective view, [34], [35] design a DETR [36], [37]-based detection head 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 [5], [38], [39], [6]. BEVDet [40] and M^{2}BEV [41] extends LSS [6] and OFT [38] to 3D object detection and CaDDN [42] adds explicit depth estimation supervision to the view transformer. Recent research [43], [8] also studies view transformation with multi-head attention.

Multi-Sensor Fusion. Recently, multi-sensor fusion arouses significant interest among the 3D detection community. Existing approaches can be classified into *proposal-level* and *point-level* fusion methods. Early approach MV3D [44] creates object proposals in 3D and projects the proposals to images to extract RoI features. [45], [46], [47] all lift image proposals into a 3D frustum. Recent work FUTR3D [48] and TransFusion [49] define object queries in the 3D space and fuses image features onto these proposals. All 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 these methods, [1], [2], [4], [50], [51] are (LiDAR) input-level decoration, while DCF [52] and DeepFusion [3] are feature-level decoration.

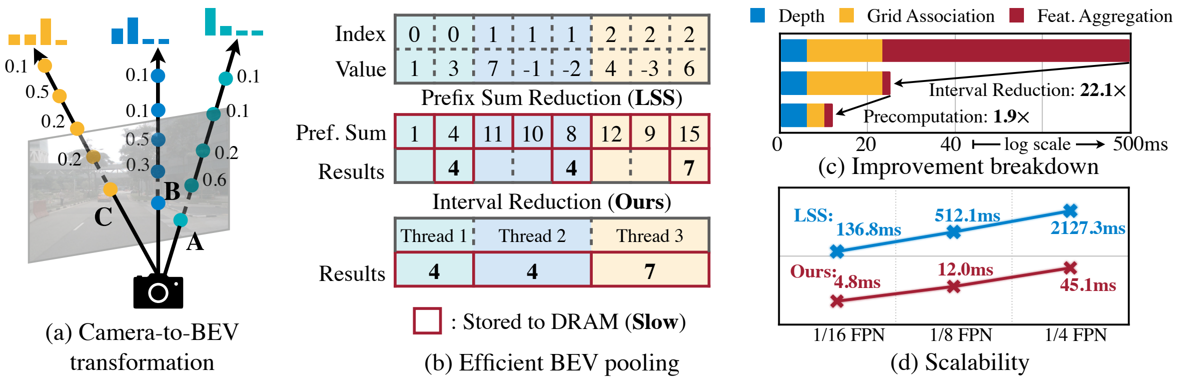
In contrast to all existing methods, BEVFusion performs sensor fusion in a shared BEV space and treats foreground and background, geometric and semantic information equally. It is a generic multi-task multi-sensor perception framework.

III. Method

BEVFusion, as shown in Figure 2, focuses on *multi-sensor fusion* (*i.e.*, multi-view cameras and LiDAR) for *multi-task 3D perception* (*i.e.*, detection and segmentation). 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.

A. Unified Representation

Different features can exist in different views. For instance, camera features are in the perspective view, while



(a) Camera-to-BEV transformation

(b) Efficient BEV pooling

(c) Improvement breakdown

(d) Scalability

Fig. 3: Camera-to-BEV transformation (a) is the key step to perform sensor fusion in the unified BEV space. Existing implementation is extremely slow and takes 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 view transformation (c, d).

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* makes the feature fusion difficult since the same element in different feature tensors might correspond to very different spatial locations (and the naïve elementwise feature fusion will not work in this case). Thus, 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 [1], [4], [3] 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) [48], [49].

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.

B. 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 [6], 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.

*$N = 6$, $(H, W) = (32, 88)$, 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×). The depth is discretized into $[1, 60]$ meters with a step size of 0.5 meter.

Tab. I: 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; †: with test-time augmentation)

	Modality	mAP (*test*)	NDS (*test*)	mAP (*val*)	NDS (*val*)	MACs (G)	Latency (ms)
M^{2}BEV [41]	C	42.9	47.4	41.7	47.0	–	–
BEVFormer [43]	C	44.5	53.5	41.6	51.7	–	–
PointPillars [10]	L	–	–	52.3	61.3	65.5	34.4
SECOND [11]	L	52.8	63.3	52.6	63.0	85.0	69.8
CenterPoint [17]	L	60.3	67.3	59.6	66.8	153.5	80.7
PointPainting [1]	C+L	–	–	65.8^{*}	69.6^{*}	370.0	185.8
PointAugmenting [2]	C+L	$66.8^{\dagger}$	$71.0^{\dagger}$	–	–	408.5	234.4
MVP [4]	C+L	66.4	70.5	66.1^{*}	70.0^{*}	371.7	187.1
FusionPainting [50]	C+L	68.1	71.6	66.5	70.7	–	–
AutoAlign [51]	C+L	–	–	66.6	71.1	–	–
FUTR3D [48]	C+L	–	–	64.5	68.3	1069.0	321.4
TransFusion [49]	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**

Tab. II: BEVFusion achieves state-of-the-art 3D object detection performance among all submissions on Waymo open dataset (`test`). (†: with test-time augmentation, ‡: with both test-time augmentation and model ensemble)

	Frames	mAP/L1	mAPH/L1	mAP/L2	mAPH/L2
AFDetV2-Ens [18]‡	**3**	84.1	82.6	79.0	77.6
InceptionLiDAR	10	83.8	82.5	79.2	77.8
3DAL-Ens [20]	5	84.6	83.1	79.7	78.2
DeepFusion-Ens [3]‡	5	84.4	83.2	79.5	78.4
MT-Net‡ [55]	3	84.7	83.2	79.9	78.5
MT3D	4	85.0	83.7	80.1	78.7
LIVOX-Detection	7	84.8	83.5	80.2	79.0
MPPNet-Ens‡ [56]	16	85.0	83.7	80.5	79.1
3DAM-Ens	5	85.3	83.8	80.7	79.2
BEVFusion (Ours)†	**3**	**85.7**	**84.4**	**80.8**	**79.5**

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 to *aggregate* the features within each BEV grid by some symmetric function (*e.g.*, mean, max, and sum). As in Figure 3b, existing implementation [6] 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 [41] or truncating the points within each BEV grid [40]. In contrast, our techniques are *exact* without any approximation, while still being faster.

C. 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 [42], [53]), which we leave for future work.

D. Multi-Task Heads

We apply multiple task-specific heads to the fused BEV feature map. Our method is applicable to most 3D perception tasks. For 3D object detection, we follow [17], [49] to 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. For map 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 [8] to train the segmentation head with the standard focal loss [54].

IV. 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 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).

Tab. III: 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 [38]	C	74.0	35.3	45.9	27.5	35.9	33.9	42.1
LSS [6]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT [8]	C	74.3	36.8	39.9	25.8	35.0	29.4	40.2
M^2BEV [41]	C	77.2	–	–	–	–	40.5	–
BEVFusion (Ours)	C	**81.7**	**54.8**	**58.4**	**47.4**	**50.7**	**46.4**	**56.6**
PointPillars [10]	L	72.0	43.1	53.1	29.7	27.7	37.5	43.8
CenterPoint [17]	L	75.6	48.4	57.5	36.5	31.7	41.9	48.6
PointPainting [1]	C+L	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP [4]	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**

Tab. IV: BEVFusion is robust under different lighting and weather conditions, significantly boosting the performance single-modality models under challenging rainy$_{(+10.7)}$ and nighttime$_{(+12.8)}$ scenes.

		Sunny		Rainy		Day		Night	
	Modality	mAP	mIoU	mAP	mIoU	mAP	mIoU	mAP	mIoU
CenterPoint [17]	L	62.9	50.7	59.2	42.3	62.8	48.9	35.4	37.0
BEVFormer [43]	C	41.0	–	44.0	–	41.9	–	21.2	–
BEVFusion	C	–	59.0	–	50.5	–	57.4	–	30.8
MVP	C+L	65.9 (+3.0)	51.0 (+0.3)	66.3 (+7.1)	42.9 (+0.6)	66.3 (+3.5)	49.2 (+0.3)	38.4 (+3.0)	37.5 (+0.5)
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)

Model. We use Swin-T [57] as our image backbone and VoxelNet [11] as our LiDAR backbone. We apply FPN [58] 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.

Dataset. We evaluate our method on nuScenes [59] and Waymo [60], which are large-scale datasets for 3D perception with >40k annotated scenes. Each sample in both datasets are equipped with both LiDAR and surrounding camera inputs.

A. 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. 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.

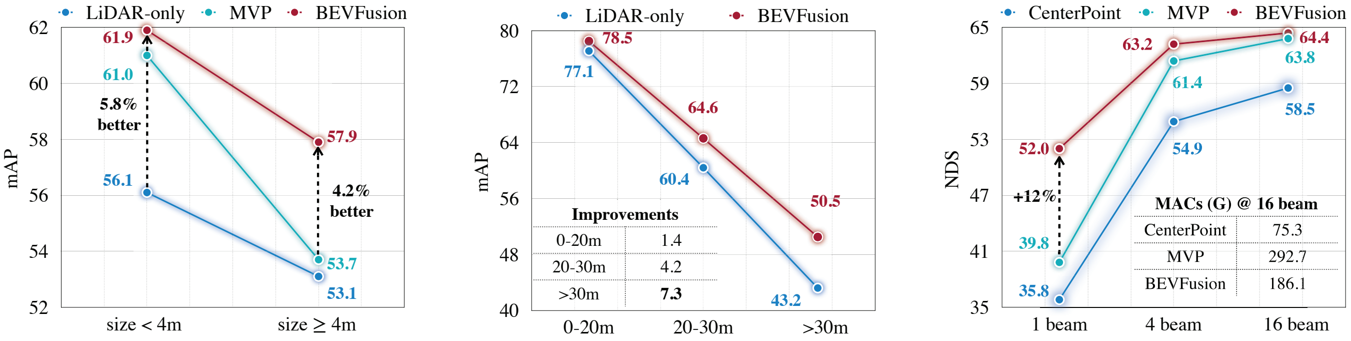
As in Table I, BEVFusion achieves state-of-the-art results on the nuScenes detection benchmark, with close-to-real-time (**8.4 FPS**) inference speed on a desktop GPU. Compared with TransFusion [49], BEVFusion offers 1.3% improvement in `test` split mAP and NDS, while significantly reduces the MACs by **1.9**× and measured latency by **1.3**×. It also compares favorably against representative point-level fusion methods PointPainting [1] and MVP [4] 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 shared fusion space, which fully utilizes all camera features instead of just a 5% sparse set. Consequently, BEVFusion can achieve the same performance with much smaller resolution for the camera inputs, resulting in significantly lower MACs. Combined with the efficient BEV pooling operator in Section III-B, BEVFusion transfers MACs reduction into measured speedup.

BEVFusion also achieves state-of-the-art performance on the Waymo open dataset [60] (Table II). BEVFusion outperforms the previous state-of-the-art multi-modal detector, DeepFusion [3] with 60% of input frames. Furthermore, DeepFusion ensembles 25 models evaluated with test-time augmentation, while we deliver better performance by applying test-time augmentation to a single BEVFusion model.

B. BEV Map Segmentation

We further compare BEVFusion with state-of-the-art models on the semantic-centric BEV map segmentation task, where BEVFusion achieves an even larger performance boost. We report the Intersection-over-Union (IoU) on 6 background classes 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 [8]. For each frame, we only perform the evaluation in the [-50m, 50m]×[-50m, 50m] region around the ego car following [6], [8], [41], [43].

We report the BEV map segmentation results in Table III. In contrast to 3D object detection which is a *geometric*-oriented task, map segmentation is *semantic*-oriented. As a result,



(a) Different object sizes

(b) Different object distances

(c) Different LiDAR sparsity

Fig. 4: BEVFusion outperforms state-of-the-art single- and multi-modality detectors under different LiDAR sparsity, object sizes and object distances, especially under more challenging settings (*i.e.*, *sparser point clouds*, *small/distant objects*).

our camera-only BEVFusion model outperforms LiDAR-only baselines by **8-13%**. This observation is the exact opposite of results in Table I, 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 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 [1], [4]. This is because both baseline methods are *object*-centric and *geometric*-oriented. PointPainting [1] 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 III.

V. Analysis

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

Weather and Lighting. We first analyze the performance of BEVFusion under different weather and lighting conditions in Table IV. LiDAR-only models face significant challenges in detecting objects in rainy weather due to sensor noise, while BEVFusion leverages the robustness of camera sensors to achieve a **10.7** mAP improvement, which largely narrows the performance gap between sunny and rainy scenarios. Poor lighting conditions pose challenges for both detection and segmentation models. For detection, MVP's improvement is relatively small compared to BEVFusion, which relies less on *accurate* 2D instance segmentations to generate virtual points and therefore performs better in dark or overexposed scenes. For segmentation, while camera-only BEVFusion outperforms CenterPoint on the entire benchmark, its performance drops significantly at nighttime. However, multi-modal BEVFusion achieves a **12.8** mIoU improvement, even greater than its improvement in the daytime, demonstrating the importance of leveraging geometric clues when camera sensors fail.

Sizes and Distances. We also analyze the performance of BEVFusion under different object sizes and distances. From Figure 4a, 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 augmented multi-modal virtual points (MVPs). Additionally, BEVFusion yields greater improvements to the LiDAR-only detector for smaller objects (in Figure 4a) and more distant objects (in Figure 4b), both of which are inadequately captured by LiDAR and can therefore derive more benefit from the dense camera information.

Sparser LiDARs. We finally demonstrate the performance of CenterPoint [17] (LiDAR-only), MVP [4] (multi-modal), and our BEVFusion under different LiDAR sparsities in Figure 4c. BEVFusion consistently outperforms MVP across all sparsity levels with a **1.6**× reduction in #MACs and achieves a **12%** improvement in the 1-beam LiDAR scenario. MVP decorates the point cloud and directly applies CenterPoint on the painted and densified LiDAR input. As a result, it naturally requires the LiDAR-only detector (CenterPoint) to perform well, which is not valid under sparse LiDAR settings (*i.e.*, 35.8 NDS with 1-beam input in Figure 4c). In contrast, BEVFusion integrates multi-sensor information in the shared BEV space and does not rely solely on a robust LiDAR-only detector.

VI. 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 geometric and semantic information. To achieve this, we accelerate the slow camera-to-BEV transformation by more than 40 times. BEVFusion rethinks the effectiveness of point-level fusion in multi-sensor perception systems and achieves superior performance on both nuScenes 3D detection and BEV map segmentation tasks with 1.5-1.9× less computation and 1.3-1.6× measured speedup over existing solutions. BEVFusion also outperforms all existing sensor fusion methods on Waymo open dataset. We hope that BEVFusion can serve as a simple but powerful baseline to inspire future research on multi-task multi-sensor fusion.

References

[1] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential Fusion for 3D Object Detection," in *CVPR*, 2020.

[2] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-Modal Augmentation for 3D Object Detection," in *CVPR*, 2021.

[3] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, B. Wu, Y. Lu, D. Zhou *et al.*, "DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection," in *CVPR*, 2022.

[4] T. Yin, X. Zhou, and P. Krähenbühl, "Multimodal Virtual Point 3D Detection," in *NeurIPS*, 2021.

[5] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-View Semantic Segmentation for Sensing Surroundings," *RA-L*, 2020.

[6] J. Philion and S. Fidler, "Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D," in *ECCV*, 2020.

[7] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HDMapNet: An Online HD Map Construction and Evaluation Framework," in *ICRA*, 2022.

[8] B. Zhou and P. Krähenbühl, "Cross-View Transformers for Real-Time Map-View Semantic Segmentation," in *CVPR*, 2022.

[9] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *CVPR*, 2018.

[10] A. H. Lang, S. Vora, H. Caesar, L. Zhou, and J. Yang, "PointPillars: Fast Encoders for Object Detection from Point Clouds," in *CVPR*, 2019.

[11] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, 2018.

[12] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-Balanced Grouping and Sampling for Point Cloud 3D Object Detection," *arXiv*, 2019.

[13] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," *CVPR*, 2020.

[14] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds," *CoRL*, 2019.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *NeurIPS*, 2017.

[16] B. Graham, M. Engelcke, and L. van der Maaten, "3D Semantic Segmentation With Submanifold Sparse Convolutional Networks," in *CVPR*, 2018.

[17] T. Yin, X. Zhou, and P. Krähenbühl, "Center-Based 3D Object Detection and Tracking," in *CVPR*, 2021.

[18] R. Ge, Z. Ding, Y. Hu, W. Shao, L. Huang, K. Li, and Q. 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.

[19] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. Yuille, "Object as Hotspots: An Anchor-Free 3D Object Detection Approach via Firing of Hotspots," in *ECCV*, 2020.

[20] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov, "Offboard 3D Object Detection from Point Cloud Sequences," in *CVPR*, 2021.

[21] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection," in *ICCV*, 2021.

[22] Q. Chen, S. Vora, and O. Beijbom, "PolarStream: Streaming Lidar Object Detection and Segmentation with Polar Pillars," in *NeurIPS*, 2021.

[23] Y. Wang and J. M. Solomon, "Object DGCNN: 3D Object Detection using Dynamic Graphs," in *NeurIPS*, 2021.

[24] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in *CVPR*, 2019.

[25] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast Point R-CNN," in *ICCV*, 2019.

[26] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network," *TPAMI*, 2020.

[27] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," in *CVPR*, 2020.

[28] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," *arXiv*, 2021.

[29] Z. Li, F. Wang, and N. Wang, "LiDAR R-CNN: An Efficient and Universal 3D Object Detector," *CVPR*, 2021.

[30] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection," in *ICCVW*, 2021.

[31] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," in *ICCV*, 2019.

[32] T. Wang, X. Zhu, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *CoRL*, 2021.

[33] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li, "EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation," in *CVPR*, 2022.

[34] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. M. Solomon, "DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries," in *CoRL*, 2021.

[35] Y. Liu, T. Wang, X. Zhang, and J. Sun, "PETR: Position Embedding Transformation for Multi-View 3D Object Detection," *arXiv*, 2022.

[36] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *ICLR*, 2021.

[37] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query Design for Transformer-Based Detector," in *AAAI*, 2022.

[38] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic Feature Transform for Monocular 3D Object Detection," in *BMVC*, 2019.

[39] T. Roddick and R. Cipolla, "Predicting Semantic Map Representations from Images using Pyramid Occupancy Networks," in *CVPR*, 2020.

[40] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View," *arXiv*, 2021.

[41] E. Xie, Z. Yu, D. Zhou, J. Philion, A. Anandkumar, S. Fidler, P. Luo, and J. M. Alvarez, "M^{2}BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation," *arXiv*, 2022.

[42] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distributionnetwork for monocular 3d object detection," in *CVPR*, 2021.

[43] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers," *arXiv*, 2022.

[44] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in *CVPR*, 2017.

[45] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *CVPR*, 2018.

[46] Z. Wang and K. Jia, "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection," in *IROS*, 2019.

[47] R. Nabati and H. Qi, "CenterFusion: Center-Based Radar and Camera Fusion for 3D Object Detection," in *WACV*, 2021.

[48] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "FUTR3D: A Unified Sensor Fusion Framework for 3D Detection," *arXiv*, 2022.

[49] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers," in *CVPR*, 2022.

[50] S. Xu, D. Zhou, J. Fang, J. Yin, B. Zhou, and L. Zhang, "FusionPainting: Multimodal Fusion with Adaptive Attention for 3D Object Detection," in *ITSC*, 2021.

[51] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, F. Zhao, B. Zhou, and H. Zhao, "AutoAlign: Pixel-Instance Feature Aggregation for Multi-Modal 3D Object Detection," *arXiv*, 2022.

[52] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep Continuous Fusion for Multi-Sensor 3D Object Detection," in *ECCV*, 2018.

[53] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is Pseudo-Lidar needed for Monocular 3D Object detection?" in *ICCV*, 2021.

[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *ICCV*, 2017.

[55] S. Chen, Z. Jie, X. Wei, and L. Ma, "MT-Net Submission to the Waymo 3D Detection Leaderboard," *arXiv*, 2022.

[56] X. Chen, S. Shi, B. Zhu, K. C. Cheung, H. Xu, and H. Li, "MPPNet: Multi-Frame Feature Intertwining with Proxy Points for 3D Temporal Object Detection," in *ECCV*, 2022.

[57] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *ICCV*, 2021.

[58] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *CVPR*, 2017.

[59] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *CVPR*, 2020.

[60] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi,

Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in Perception for Autonomous Driving: Waymo Open Dataset,” in *CVPR*.