Delving Into the Devils of Bird's-Eye-View Perception: A Review, Evaluation and Recipe

Hongyang Li, *Senior Member, IEEE*, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, Hao Tian, Enze Xie, Jiangwei Xie, Li Chen, Tianyu Li, Yang Li, Yulu Gao, Xiaosong Jia, Si Liu, Jianping Shi, *Member, IEEE*, Dahua Lin, and Yu Qiao, *Senior Member, IEEE*

(Survey Paper)

***Abstract*—Learning powerful representations in bird's-eye-view (BEV) for perception tasks is trending and drawing extensive attention both from industry and academia. Conventional approaches for most autonomous driving algorithms perform detection, segmentation, tracking, etc., in a front or perspective view. As sensor configurations get more complex, integrating multi-source information from different sensors and representing features in a unified view come of vital importance. BEV perception inherits several advantages, as representing surrounding scenes in BEV is intuitive and fusion-friendly; and representing objects in BEV is most desirable for subsequent modules as in planning and/or control. The core problems for BEV perception lie in (a) how to reconstruct the lost 3D information via view transformation from perspective view to BEV; (b) how to acquire ground truth annotations in BEV grid; (c) how to formulate the pipeline to incorporate features from different sources and views; and (d) how to adapt and generalize algorithms as sensor configurations vary across different scenarios. In this survey, we review the most recent works on BEV perception and provide an in-depth analysis of different solutions. Moreover, several systematic designs of BEV approach from the industry are depicted as well. Furthermore, we introduce a full suite of practical guidebook to improve the performance of BEV perception tasks, including camera, LiDAR and fusion inputs. At last, we point out the future research directions in this area. We hope this report will shed some light on the community and encourage more research effort on BEV perception.**

***Index Terms*—3D detection and segmentation, autonomous driving challenge, bird 's-eye-view (BEV) perception.**

Manuscript received 13 September 2022; revised 15 September 2023; accepted 4 November 2023. Date of publication 17 November 2023; date of current version 6 March 2024. This work was supported by the National Key R&D Program of China under Grant 2022ZD0160104, and in part by the NSFC under Grant 62206172. Recommended for acceptance by D. Crandall. *(Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, Hao Tian, and Enze Xie contributed equally to this work.) (Corresponding author: Hongyang Li.)*

Hongyang Li, Jifeng Dai, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Tianyu Li, Yang Li, Xiaosong Jia, Dahua Lin, and Yu Qiao are with Shanghai Artificial Intelligence Laboratory, Shanghai 201203, China (e-mail: hongyangli2020@gmail.com; daijifeng@tsinghua.edu.cn; huijie.wang@hotmail.com; jia.zeng@sjtu.edu.cn; lzq@smail.nju.edu.cn; ytepjz@gmail.com; litianyu@pjlab.org.cn; ricardleedhu@gmail.com; jiaxiaosong@sjtu.edu.cn; dhlin@ie.cuhk.edu.hk; yu.qiao@siat.ac.cn).

Lewei Lu, Hanming Deng, Hao Tian, Jiangwei Xie, and Jianping Shi are with SenseTime, Beijing 200233, China (e-mail: luotto@sensetime.com; denghanming@sensetime.com; tianhaohust@gmail.com; xiejw@shanghaitech.edu.cn; shijianping@sensetime.com).

Chonghao Sima and Li Chen are with Shanghai Artificial Intelligence Laboratory, Shanghai 201203, China, and also with the University of Hong Kong, Hong Kong (e-mail: chonghaosima@gmail.com; ilnehc@umich.edu).

Wenhai Wang is with the Chinese University of Hong Kong, Hong Kong (e-mail: wangwenhai362@163.com).

Enze Xie is with the Huawei Inc., Shenzhen 518129, China (e-mail: xieenze@connect.hku.hk).

Yulu Gao and Si Liu are with the Beihang University, Beijing 100191, China (e-mail: gyl97@buaa.edu.cn; liusi@buaa.edu.cn).

We keep an active repository to collect the most recent work and provide a toolbox for bag of tricks at https://github.com/OpenDriveLab/Birds-eye-view-Perception.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3333838, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3333838

I. Introduction

PERCEPTION recognition task in autonomous driving is essentially a 3D geometry reconstruction to the physical world. As the diversity and number of sensors become more and more complicated in equipping the self-driving vehicle (SDV), representing features from different views in a unified perspective comes to vital importance. The well-known bird's-eye-view (BEV) is a natural and straightforward candidate view to serve as a unified representation. Compared to its front-view or perspective view counterpart, which is broadly investigated [1], [2] in 2D vision domains, BEV representation bears several inherent merits. First, it has no occlusion or scale problem as commonly existent in 2D tasks. Recognizing vehicles with occlusion or cross-traffic could be better resolved. Moreover, representing objects or road elements in such form would favorably make it convenient for subsequent modules (e.g. planning, control) to develop and deploy.

In this survey, we term **BEV Perception** to indicate all vision algorithms in sense of the BEV view representation for autonomous driving. Note that we do not intend to exaggerate BEV perception as a new research concept; instead, how to formulate new pipeline or framework under BEV view for better feature fusion from multiple sensor inputs, deserves more attention from the community.

A. Big Picture At a Glance

Based on the input data, we divide BEV perception research into three parts mainly - BEV camera, BEV LiDAR and BEV

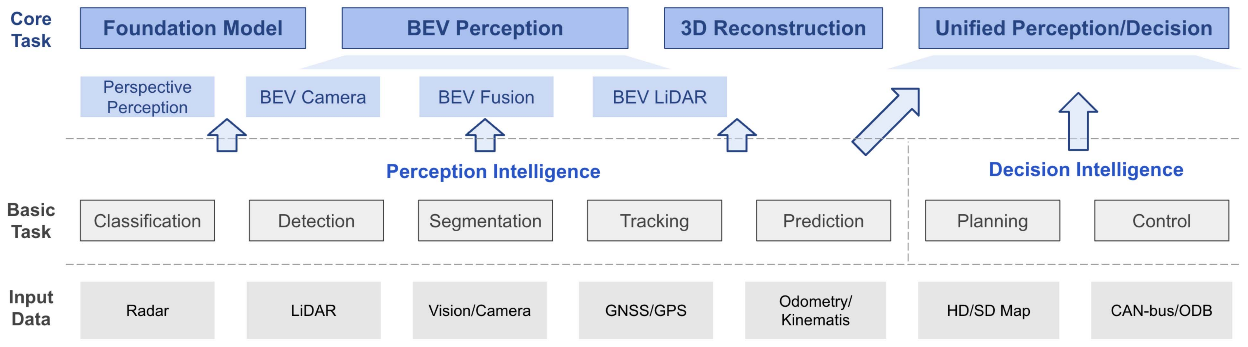


Fig. 1. General picture of BEV perception at a glance, where consists of three sub-parts based on the input modality. BEV perception is a general task built on top of a series of fundamental tasks. For better completeness of the whole perception algorithms in autonomous driving, we list other topics (e.g., Foundation Model) as well.

fusion. Fig. 1 depicts the general picture of BEV perception family. Specifically, **BEV camera** indicates vision-only or vision-centric algorithms for 3D object detection or segmentation from multiple surrounding cameras; **BEV LiDAR** describes detection or segmentation task from point cloud input; **BEV fusion** depicts the fusion mechanisms from multiple sensor inputs, such as camera, LiDAR, GNSS, odometry, HD-Map, CAN-bus, etc.

As described in Fig. 1, we group and classify fundamental perception algorithms (classification, detection, segmentation, tracking, etc.) with autonomous driving tasks into three levels, where the concept of BEV perception lies in the middle. Based on different combinations from the sensor input layer, fundamental task, and product scenario, a certain BEV perception algorithm could indicate accordingly. For instance, M^2 BEV [3] and BEVFormer [4] belonged to BEV camera track from multiple cameras to perform multiple tasks including 3D object detection and BEV map segmentation. BEVFusion [5] devised a fusion strategy in BEV space to perform 3D detection and tracking simultaneously from both camera and LiDAR inputs. Tesla [6] released its systematic pipeline to detect objects and lane lines in vector space (BEV) for L2 highway navigation and smart summon. In this report, we aim to summarize a general pipeline and key insights for recent advanced BEV perception research, apart from various input combinations and tasks.

B. Motivation to BEV Perception Research

When it comes to the motivation of BEV perception research, three important aspects need to be examined.

Significance: Will BEV perception generate a real and meaningful impact on academia and/or society? It is well-known that a tremendous performance gap exists between camera-only and LiDAR solutions. For example, as of submission in August 2022, the gap between first-ranking camera-only and LiDAR methods exceed 20% on nuScenes dataset [7] and 30% on Waymo benchmark [8].This naturally prompts us to investigate whether the camera-only solutions can beat or be on par with the LiDAR approaches.

From the academic perspective, the very essence of designing a camera-based pipeline such that it outperforms LiDAR, is better to understand the view transformation process from a 2D, appearance input to a 3D, geometry output. How to transfer camera features into geometry representations as does in point cloud leaves a meaningful impact for the academic society. On the industrial consideration, the cost of a suite of LiDAR equipment into SDV is expensive; OEMs (Original Equipment Manufacturer, e.g., Ford, BMW, etc.) prefer a cheap as well as accurate deployment for software algorithms. Improving camera-only algorithms to LiDAR's naturally fall into this goal since the cost of a camera is usually 10x times cheaper than LiDAR. Moreover, camera based pipeline can recognize long-range distance objects and color-based road elements (e.g., traffic lights), both of which LiDAR approaches are incapable of.

Although there are several different solutions for camera and LiDAR based perception, BEV representation is one of the best candidates for LiDAR based methods in terms of superior performance and industry-friendly deployment. Moreover, recent trends show that BEV representation also has huge progress for multi-camera inputs. Since camera and LiDAR data can be projected to BEV space, another potential for BEV is that we can easily fuse the feature from different modalities under a unified representation.

Space: Are there open problems or caveats in BEV perception that require substantial innovation? The gist behind BEV perception is to learn a robust and generalizable feature representation from both camera and LiDAR inputs. This is easy in LiDAR branch as the input (point cloud) bears such a 3D property. This is non-trivial otherwise in the camera branch, as learning 3D spatial information from monocular or multi-view settings is difficult. While we see there are some attempt to learn better 2D-3D correspondence by pose estimation [9] or temporal motion [10], the core issue behind BEV perception requires a substantial innovation of depth estimation from raw sensor inputs, esp. for the camera branch.

Another key problem is how to fuse features in early or middle stage of the pipeline. Most sensor fusion algorithms treat the problem as a simple object-level fusion or naive feature concatenation along the blob channel. This might explain why some fusion algorithms behave inferior to LiDAR-only solutions, due to the misalignment or inaccurate depth prediction between camera and LiDAR. How to align and integrate features from multi-modality input plays a vital role and thereby leaves extensive space to innovate.

Readiness: Are key conditions (e.g., dataset, benchmark) ready to conduct BEV perception research? The short answer is yes. As BEV perception requires both camera and LiDAR, high-quality annotation and accurate alignment between 2D and 3D objects are two key evaluations for such benchmarks. While KITTI [11] is comprehensive and draws much attention in early autonomous driving research, large-scale and diverse benchmarks such as Waymo [8], nuScenes [7], Argoverse [12] provide a solid playground for validating BEV perception ideas. These newly proposed benchmarks usually feature high-quality labels; the scenario diversity and data amount also scale up to great extent. Furthermore, open challenges [13] on these leaderboards give a fair setting on the held-out test data, where all state-of-the-arts can be compared in an open and prompt sense.

As for the algorithm readiness, recent years have witnessed a great blossom for general vision, where Transformer [14], ViT [15], [16], Masked Auto-encoders (MAE) [17] and CLIP [18], etc., achieve impressive gain over conventional methods. We believe these work would benefit and inspire greatness for of BEV perception research.

Based on the discussions of the three aspects above, we conclude that BEV perception research has great potential impact, and deserves massive attention from both academia and industry to put great effort for a long period. Compared to the recent surveys on 3D object detection [19], [20], [21], [22], [23], our survey not only summarizes recent BEV perception algorithms at a more high level and formulates them into a general pipeline, but also provides useful recipe under such context, including solid data augmentation in both camera-based and LiDAR-based settings, efficient BEV encoder design, perception heads and loss function family, useful test-time augmentation (TTA) and ensemble policy, and so on. We hope this survey can be a good starting point for new beginners and an insightful discussion for current researchers in this community.

C. Contributions

The main contributions in this survey are three-fold:

1) We review the whole picture of BEV perception research in recent years, including high-level philosophy and in-depth detailed discussion.
2) We elaborate on a comprehensive analysis of BEV perception literature. The core issues such as depth estimation, view transformation, sensor fusion, domain adaptation, etc., are covered. Several important industrial system-level designs for BEV perception are introduced and discussed as well.
3) We provide a practical guidebook, besides theoretical contribution, for improving the performance in various BEV perception tasks. Such a release could facilitate the community to achieve better performance in a grab-and-go recipe sense.

II. Background in 3D Perception

In this section, we present the essential background knowledge in 3D perception. In Section II-A, we review conventional approaches to perform perception tasks, including monocular camera based 3D object detection, LiDAR based 3D object detection and segmentation, and sensor fusion strategies. In Section II-B, we introduce predominant datasets in 3D perception, such as KITTI dataset [11], nuScenes dataset [7] and Waymo Open Dataset [8].

A. Task Definition and Related Work

Monocular Camera-based Object Detection: Monocular camera-based methods take an RGB image as input and attempt to predict the 3D location and category of each object. The main challenge of monocular 3D detection is that the RGB image lacks depth information, so these kinds of methods need to predict the depth. Due to estimating depth from a single image being an ill-posed problem, typically monocular camearbased methods have inferior performance than LiDAR-based methods.

LiDAR Detection and Segmentation: LiDAR describes surrounding environments with a set of points in the 3D space, which capture geometry information of objects. Despite the lack of color and texture information and the limited perception range, LiDAR-based methods outperform camera-based methods by a large margin due to the depth prior.

Sensor Fusion: Modern autonomous vehicles are equipped with different sensors such as cameras, LiDAR and Radar. Each sensor has advantages and disadvantages. Camera data contains dense color and texture information but fails to capture depth information. LiDAR provides accurate depth and structure information but suffers from limited range and sparsity. Radar is more sparse than LiDAR, but has a longer sensing range and can capture information from moving objects. Ideally, sensor fusion would push the upper bound performance of the perception system, and yet how to fuse data from different modalities remains a challenging problem.

B. Datasets and Metrics

We introduce some popular autonomous driving datasets and the common evaluation metrics. Table I summarizes the main statistics of prevailing benchmarks for BEV perception. Normally, a dataset consists of various scenes, each of which is of different length in different datasets. The total duration ranges from tens of minutes to hundreds of hours. For BEV perception tasks, 3D bounding box annotation and 3D segmentation annotation are essential and HD-Map configuration has become a mainstream trend. Most of them can be adopted in different tasks. A consensus reached that sensors with multiple modalities and various annotations are required. More types of data [7], [12], [24], [25], [33], [39] are released such as IMU/GPS and CAN-bus. Similar to the Kaggle and EvalAI leaderboards, we reveal the total number of submissions on each dataset to indicate the popularity of a certain dataset.

1) Datasets. KITTI Dataset: KITTI [11] is a pioneering autonomous driving dataset proposed in 2012. It has 7481 training images and 7518 test images for 3D object detection tasks. It also has corresponding point clouds captured from the Velodyne laser scanner. The test set is split into 3 parts: easy,

TABLE I
BEV PERCEPTION DATASETS AT A GLANCE

Dataset	Year	Region	Sensor Data				Annotation				HD-Map	Other Data	# Subm.
			Scenes	Hours	Scans	Images	Frames	3D bbox	3D lane	3D seg.			
KITTI [11]	2012	EU	22	1.5	15k	15k	15k	80k	-	-	✗	-	380+
Waymo [8]	2019	NA	1150	6.4	230k	12M	230k	12M	-	50k	✗	-	200+
nuScenes [7]	2019	NA/AS	1000	5.5	390k	1.4M	40k	1.4M	-	40k	✓	CAN-bus	350+
Argo v1 [24]	2019	NA	113	0.6	22k	490k	22k	993k	-	-	✓	-	100+
Argo v2 [12]	2022	NA	1000	4	150k	2.7M	150k	†	-	-	✓	-	10+
ApolloScape [25]	2018	AS	103	2.5	29k	144k	144k	70k	†	-	✓	-	200+
OpenLane [26]	2022	NA	1000	6.4	-	200k	200k	-	880k	-	✗	-	-
ONCE-3DLanes [27]	2022	AS	†	†	-	211k	211k	-	†	-	✗	-	-
Lyft L5 [28]	2019	AF	366	2.5	46k	240k	46k	1.3M	-	-	✗	-	500+
A* 3D [29]	2019	AS	†	55	39k	39k	39k	230k	-	-	✗	-	-
H3D [30]	2019	NA	160	0.8	27k	83k	27k	1.1M	-	-	✗	-	-
SemanticKITTI [31]	2019	EU	22	1.2	43k	-	-	-	-	43k	✗	-	30+
A2D2 [32]	2020	EU	†	†	12.5k	41k	12.5k	43k	-	41k	✗	-	-
Cityscapes 3D [33]	2020	-	†	2.5	-	5k	5k	40k	-	-	✗	IMU/GPS	400+
PandaSet [34]	2020	NA	179	†	16k	41k	14k	†	-	60k	✗	-	-
KITTI-360 [35]	2020	EU	11	†	80k	320k	80k	68k	-	80k	✗	-	30+
Cirrus [36]	2020	-	12	†	6285	6285	6285	†	-	-	✗	-	-
ONCE [37]	2021	AS	†	144	1M	7M	15k	417k	-	-	✗	-	-
AIODrive [38]	2021	Sim	100	2.8	100k	1M	100k	26M	-	100k	✓	-	-
DeepAccident [39]	2022	Sim	464	†	131k	786k	131k	1.8M	-	131k	✓	-	-

Scenes indicates clips of dataset, and the length of scene is varying for diverse datasets. Under Region, "AS" for Asia, "EU" for Europe, "NA" for North America, "Sim" for simulation data. Under Sensor Data, "Scans" for point cloud. Under Annotation, Frames implies the number of 3D bbox/ 3D lane annotation frames, 3D bbox/ 3D lane represents the number of 3D bbox/ 3D lane annotation instances, 3D seg. means the number of segmentation annotation frames of point cloud. "# Subm." denotes the popularity of a particular dataset by the number of submissions on Kaggle. † refers that the statistic is unavailable; − means that the field is non-existent.

moderate, and hard, mainly by the bounding box size and the occlusion level. The evaluation of object detection is of two types: 3D object detection evaluation and bird's eye view evaluation. KITTI is the first comprehensive dataset for multiple autonomous driving tasks, and it draws massive attention from the community.

Waymo Dataset: The Waymo Open Dataset v1.3 [8] contains 798, 202 and 80 video sequences in the training, validation, and testing sets, respectively. Each sequence has 5 LiDARs and 5 views of side left, front left, front, front right and side right image resolution is 1920×1280 pixels or 1920×886 pixels. Waymo is large-scale and diverse. It is evolving as the dataset version keeps updating. Each year Waymo Open Challenge would define new tasks and encourage the community to work on the problems.

nuScenes Dataset: The nuScenes Dataset [7] is a large scale autonomous driving dataset which contains 1000 driving scenes in two cities. 850 scenes are for training/validation and 150 are for testing. Each scene is 20s long. It has 40k keyframes with entire sensor suite including 6 cameras, 1 LiDAR and 5 Radars. The camera image resolution is 1600×900. Meanwhile, corresponding HD-Map and CAN-bus data are released to explore assistance of multiple inputs. nuScenes becomes more and more popular in the academic literature as it provides a diverse multi-sensor setting; the data scale is not as large as Waymo's, making it efficient to fast verify idea on this benchmark.

2) Evaluation Metrics. LET-3D-APL: In the camera-only 3D detection, LET-3D-APL is used as the metric instead of 3D-AP. Compared with the 3D intersection over union (IoU), the LET-3D-APL allows longitudinal localization errors of the predicted bounding boxes up to a given tolerance. LET-3D-APL penalizes longitudinal localization errors by scaling the precision using the localization affinity. The definition of LET-3D-APL is mathematically defined as:

$$\text{LET-3D-APL} = \int_0^1 p_L(r)dr = \int_0^1 \overline{a}_l \cdot p(r)dr, \tag{1}$$

where $p_L(r)$ indicates the longitudinal affinity weighted precision value, the $p(r)$ means the precision value at recall r, and the multiplier $\overline{a}_l$ is the average longitudinal affinity of all matched predictions treated as TP (true positive).

mAP: The mean Average Precision (mAP) is similar to the well-known AP metric in the 2D object detection, but the matching strategy is replaced from IoU to the 2D center distance on the BEV plane. The AP is calculated under different distance thresholds: 0.5, 1, 2, and 4 meters. The mAP is computed by averaging the AP in the above thresholds.

NDS: The nuScenes detection score (NDS) is a combination of several metrics: mAP, mATE (Average Translation Error), mASE (Average Scale Error), mAOE (Average Orientation Error), mAVE (Average Velocity Error) and mAAE (Average Attribute Error). The NDS is computed by using the weight-sum of the above metrics. The weight of mAP is 5 and 1 for the rest. In the first step the TP_{error} is converted to TP_{score} as shown in (2), then (3) defines the NDS:

$$\text{TP}_{\text{score}} = max(1 - \text{TP}_{\text{error}}, 0.0), \tag{2}$$

$$\text{NDS} = \frac{5 \cdot \text{mAP} + \sum_{i=1}^{5} \text{TP}^{\text{i}}_{\text{score}}}{10}. \tag{3}$$

TABLE II
BEV PERCEPTION LITERATURE IN RECENT YEARS

Method	Venue	Input Modality	Task	Depth Supervision	Dataset	Contribution
OFT [43]	*BMVC 2019*	SC	ODet	✗	KITTI	Feature Projection to BEV
VoxelNet [44]	*CVPR 2018*	L	ODet	-	KITTI	Implicit voxel grids transformed to BEV
PointPillars [45]	*CVPR 2019*	L	ODet	-	KITTI	Voxelization with pillars as BEV
CaDDN [46]	*CPVR 2021*	SC	ODet	✓	KITTI/WOD	Depth Distribution with Supervision
DfM [10]	*ECCV 2022*	SC	ODet	✓	KITTI	Motion to Depth to Voxel to BEV
BEVDet [47]	*arXiv 2022*	MC	ODet	✗	nuS	BEV space data augmentation
PETR [48]	*arXiv 2022*	MC	ODet	✗	nuS	Implicit BEV Pos Embed
BEVDepth [49]	*arXiv 2022*	MC/T	ODet	✓	nuS	Depth Correction
ImVoxelNet [50]	*WACV 2022*	SC/MC	ODet	✗	nuS/KITTI SUN/ScanNet	Camera Pose with Grid Sampler to BEV
M^{2}BEV [3]	*arXiv 2022*	MC	ODet/MapSeg	✗	nuS	BEV Representation without Depth
PolarFormer [51]	*arXiv 2022*	MC	ODet/MapSeg	✗	nuS	Polar-ray Representation in BEV
BEVFormer [4]	*ECCV 2022*	MC/T	ODet/MapSeg	✗	nuS/WOD	Transformer for BEV feature
BEVFusion [5]	*arXiv 2022*	MC/L	ODet/MapSeg	-	nuS	Fusion on BEV from Camera and LiDAR
Cam2BEV [52]	*ITSC 2020*	MC	MapSeg	✗	Synthetic	Homo-graphic Projection to BEV
FIERY [53]	*ICCV 2021*	MC	MapSeg	✗	nuS/Lyft	Future Prediction in BEV space
CVT [54]	*CVPR 2022*	MC	MapSeg	✗	nuS	Camera Intrinsics BEV Projection
HDMapNet [55]	*ICRA 2022*	MC/L/T	MapSeg	-	nuS	Feature Fusion under BEV
Image2Map [56]	*ICRA 2022*	SC/T	MapSeg	✗	nuS/Lyft/AV	Polar-ray Transformer in BEV
LSS [57]	*ECCV 2020*	MC	MapSeg/Plan	✗	nuS/Lfyt	First Depth Distribution
ST-P3 [58]	*ECCV 2022*	MC/T	MapSeg/Plan	✓	nuS/Carla	End to End P3 with Temporal Info
3D LaneNet [59]	*ICCV 2019*	SC	LDet	✗	OpenLane	IPM Projection to BEV space
STSU [60]	*ICCV 2021*	SC	LDet	✗	nuS	Query-based Centerline in BEV
PersFormer [26]	*ECCV 2022*	SC	LDet	✗	OpenLane	Perspective Transformer for BEV

Under Input Modality, "L" for LiDAR, "SC" for single camera, "MC" for multi camera, "T" for temporal information. Under Task, "ODet" for 3D object detection, "LDet" for 3D lane detection, "MapSeg" for map segmentation, "Plan" for motion planning, "MOT" for multi-object tracking. Depth Supervision means either camera-only model uses sparse/dense depth map to supervise the model, ✓ for yes, ✗ for no, - for LiDAR-input model. Under Dataset, "nuS" nuScenes dataset [7], "WOD" Waymo Open Dataset [8], "KITTI" KITTI dataset [11], "Lyft" Lyft Level 5 Dataset [28], "OpenLane" OpenLane dataset [26], "AV" Argoverse Dataset [24], "Carla" carla simulator [40], "SUN" SUN RGB-D dataset [41], "ScanNet" ScanNet indoor scenes dataset [42].

III. Methodology of BEV Perception

In this section, we describe in great details various perspectives of BEV perception from both academia and industry. We differentiate BEV pipeline in three settings based on the input modality, namely BEV Camera (camera-only 3D perception) in Section III-A, BEV LiDAR in Section III-B and BEV Fusion in Section III-C, and summarize industrial design of BEV perception in Section III-D.

Table II summarizes the taxonomy of BEV perception literature based on input data and task type. We can see that there is trending research for BEV perception published in top-tiered venues. The task topics as well as the formulation pipelines (contribution) can be various, indicating a blossom of the 3D autonomous driving community. Table III depicts the performance gain of 3D object detection and segmentation on popular leaderboards over the years. We can observe that the performance gain improves significantly in spirit of BEV perception knowledge.

A. BEV Camera

1) General Pipeline: Camera-only 3D perception attracts a large amount of attention from academia. The core issue is that 2D imaging process inherently cannot preserve 3D information, hindering precise object localization without accurate depth extraction. Camera-only 3D perception can be split into three domains: monocular setting, stereo setting and multi-camera setting. Since multi-camera methods often start with a monocular baseline, we start with monocular baseline setting as well. We use "2D space" to refer to perspective view with pixel coordinates, "3D space" to refer to 3D real-world space with world coordinates, and "BEV space" to refer to bird's eye view in the following context. As depicted in Fig. 2, a general camera-only 3D perception system can be divided into three parts: 2D feature extractor, view transformation module (optional), and 3D decoder. As camera-only 3D perception has the same input as 2D perception, the general feature extractor can be formulated as:

$$\mathcal{F}^{*}_{2D}(u,v) = M_{feat}(\mathcal{I}^{*}(u,v)), \tag{4}$$

where $\mathcal{F}_{2D}$ denotes 2D feature, $\mathcal{I}$ denotes image, M_{feat} denotes 2D feature extractor, u, v denote coordinates on 2D plane and * denotes one or more images and corresponding 2D features. In 2D feature extractor, there exists a huge amount of experience in 2D perception that can be considered in 3D perception, in the form of backbone pretraining [79], [80]. The view transformation module largely differentiates from 2D perception system. Note that not all 3D perception methods have a view transformation module, and some detect objects in 3D space directly from features in 2D space [80], [81], [82]. As depicted in Fig. 2, there are generally three ways to perform view transformation. Such a transformation can be formulated as:

$$\mathcal{F}_{3D}(x,y,z) = M_{trans}\left(\mathcal{F}^{*}_{2D}(\hat{u},\hat{v}), \begin{bmatrix}\boldsymbol{R} & \boldsymbol{T}\end{bmatrix}, \boldsymbol{K}\right), \tag{5}$$

where $\mathcal{F}_{3D}$ denotes 3D (or voxel) feature, x, y, z denote coordinates in 3D space, M_{trans} denotes view transformation module, $\hat{u}, \hat{v}$ denote corresponding 2D coordinates in terms of x, y, z, $\begin{bmatrix}\boldsymbol{R} & \boldsymbol{T}\end{bmatrix}$ and $\boldsymbol{K}$ are camera extrinsics and intrinsics as described in Section B of Appendix, available online. Note that some methods do not rely on camera extrinsics and intrinsics. The 3D decoders receive the feature in 2D/3D space and output

TABLE III
PERFORMANCE COMPARISON OF BEV PERCEPTION ALGORITHMS ON POPULAR BENCHMARKS

Methods	Modality			Task Head		KITTI ODet %			nuS ODet		nuS MapSeg		OL	WOD	
	SC	MC	L	Det	Seg	Easy	Mod.	Hard	NDS	mAP	DRI	LAN	F1	LET	APH/L2
OFT [43]	✓	-	-	✓	-	2.50	3.28	2.27	-	-	-	-	-	-	-
CaDDN [46]	✓	-	-	✓	-	19.17	13.41	11.46	-	-	-	-	-	-	-
ImVoxelNet [50]	✓	✓	-	✓	-	17.15	10.97	9.15	-	0.518*	-	-	-	-	-
DfM [10]	✓	-	-	✓	-	22.94	16.82	14.65	-	-	-	-	-	-	-
PGD [62]	✓	-	-	✓	-	19.05	11.76	9.39	0.448	0.386	-	-	-	0.5127	-
BEVerse [63]	-	✓	-	✓	-	-	-	-	0.531	0.393	-	-	-	-	-
PointPillars [45]	-	-	✓	✓	-	-	-	-	0.550	0.401	-	-	-	-	-
BEVDet [47]	-	✓	-	✓	-	-	-	-	0.552	0.422	-	-	-	-	-
BEVDet4D [64]	-	✓	-	✓	-	-	-	-	0.569	0.451	-	-	-	-	-
BEVDepth [49]	-	✓	-	✓	-	-	-	-	0.609	0.520	-	-	-	-	-
DSGN [65]	-	✓	-	✓	-	73.50	52.18	45.14	-	-	-	-	-	-	-
PV-RCNN [66]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7152
CenterPoint [67]	-	-	✓	✓	-	-	-	-	0.673	0.603	-	-	-	-	0.7193
SST [68]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7281
AFDetV2 [69]	-	-	✓	✓	-	-	-	-	0.685	0.624	-	-	-	-	0.7312
PV-RCNN++ [70]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7352
Pyramid-PV [71]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7443
MPPNet [72]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7567
M^{2}BEV [3]	-	✓	-	✓	✓	-	-	-	0.474	0.429	77.3	40.5	-	-	-
BEVFormer [4]	-	✓	-	✓	✓	-	-	-	0.569	0.481	80.1	25.7	-	0.5616	-
PolarFormer [51]	-	✓	-	✓	✓	-	-	-	0.572	0.493	82.6	46.2	-	-	-
PointPainting [73]	-	✓	✓	✓	-	-	-	-	0.581	0.464	-	-	-	-	-
MVP [74]	-	✓	✓	✓	-	-	-	-	0.705	0.664	-	-	-	-	-
AutoAlign [75]	-	✓	✓	✓	-	-	-	-	0.709	0.658	-	-	-	-	-
TransFusion [76]	-	✓	✓	✓	-	-	-	-	0.717	0.689	-	-	-	-	-
DeepFusion [77]	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	0.7554
AutoAlignV2 [78]	-	✓	✓	✓	-	-	-	-	0.724	0.684	-	-	-	-	-
BEVFusion [5]	-	✓	✓	✓	✓	-	-	-	0.761	0.750	85.5	53.7	-	-	0.7633
3D-LaneNet [59]	✓	-	-	✓	-	-	-	-	-	-	-	-	0.441	-	-
PersFormer [26]	✓	-	-	✓	-	-	-	-	-	-	-	-	0.505	-	-

We classify different approaches following Tab. 2. Under Modality, "SC," "MC," "L" denote single camera, multi camera and LiDAR, respectively. Under Task Head, "Det" designates 3D object/lane detection task, and "Seg" indicates BEV map segmentation task. Under KITTI ODet, we report AP40 of 3D object at Easy, Moderate and Hard level in KITTI dataset [11]. Under nuS ODet, we report NDS and mAP of 3D object in nuScenes dataset [7]. Under nuS MapSeg, we report mIOU score of DRI(drivable area) and LAN(lane, a.k.a divider) categories in nuScenes Map Segmentation setting. Under OL, we report F1 score of 3D laneline in OpenLane dataset [26]. Under WOD, we report LET-APL [61] for camera-only 3D object detection and APH/L2 [8] for any modality 3D object detection in Waymo Open Dataset [8]. * denotes results reported by the original papers.

3D perception results such as 3D bounding boxes, BEV map segmentation, 3D lane keypoints and so on. Most 3D decoders come from LiDAR-based methods [44], [67], [83], [84] which perform detection in voxel-space/BEV space, but there are still some camera-only 3D decoders that utilize the feature in 2D space [81], [82], [85] and directly regress the localization of 3D objects.

2) View Transformation: View transformation module is pivotal in camera-only 3D perception, as it serves as the primary unit for constructing 3D data and encoding 3D prior assumptions. Recent studies [3], [4], [10], [26], [47], [48], [49], [51], [56], [59] have centered on enhancing this module. We divide the view transformation techniques into three principal streams. The first stream, designated as "2D-3D methods", commences with 2D image features and "lifts" 2D features to 3D space via depth estimation. The second stream, recognized as "3D-2D methods", originates in 3D space, and encode 2D feature to 3D space via 3D-to-2D projection mapping. The first two stream explicitly model geometric transformation relationships. In contrast, the third stream, denoted as "pure-network-based methods", utilizes neural networks to implicitly acquire the geometric transformation. Fig. 3 presents a summary roadmap of performing view transformation, and they are analyzed in details below.

(1) *2D-3D methods:* The *2D-3D* method, firstly introduced by LSS [57], predicts grid-wise depth distribution on 2D features, then "lifts" the 2D features to voxel space based on depth, and conducts downstream tasks similar to LiDAR-based methods. This process can be formulated as:

$$\mathcal{F}_{3D}(x,y,z) = \left[\mathcal{F}^*_{2D}(\hat{u},\hat{v}) \otimes \mathcal{D}^*(\hat{u},\hat{v})\right]_{xyz}, \tag{6}$$

where $\mathcal{F}_{3D}(x,y,z)$ and $\mathcal{F}^*_{2D}(\hat{u},\hat{v})$ remain the same meaning as (5), $\mathcal{D}^*(\hat{u},\hat{v})$ represents predicted depth value or distribution at $(\hat{u},\hat{v})$, and $\otimes$ denotes outer production or similar operations. Note that this is very different from pseudo-LiDAR methods [86], [87] whose depth information is extracted from a pretrained depth estimation model and the lifting process occurs before 2D feature extraction. After LSS [57], there is another work following the same idea of formulating depth as bin-wise distribution, namely CaDDN [46]. CaDDN employs a similar network to predict categorical depth distribution, squeezes the voxel-space feature down to BEV space, and performs 3D detection at the end. The main difference between LSS [57] and CaDDN [46] is that CaDDN uses depth ground truth to supervise its categorical depth distribution prediction, thus owing a superior depth network to extract 3D information from 2D space. This track comes subsequent work such as BEVDet [47]

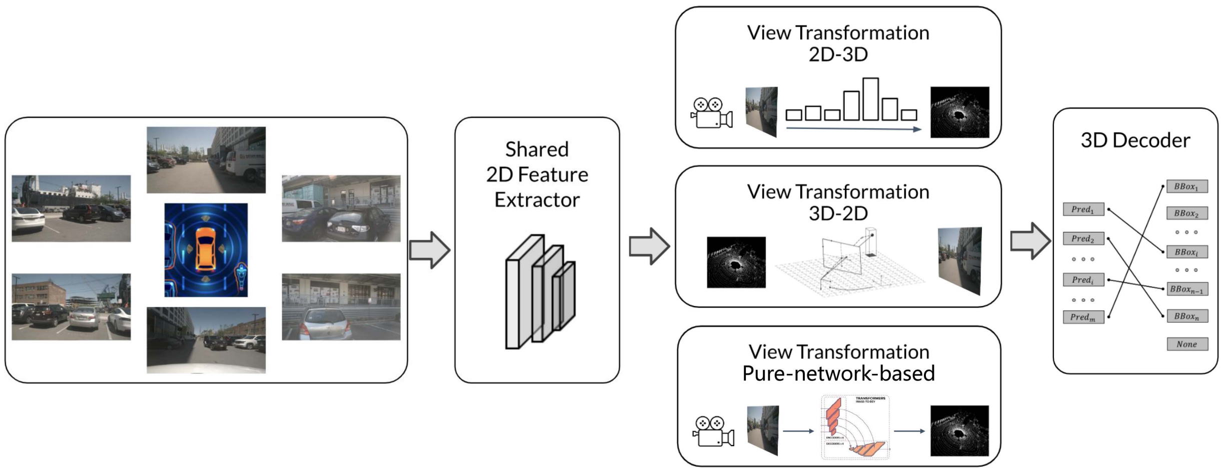


Fig. 2. General pipeline of BEV Camera (camera-only perception). There are three parts, including 2D feature extractor, view transformation and 3D decoder. In view transformation, there are three ways to encode 3D information - the first one is to predict depth information from 2D feature; the second one is to sample 2D feature from 3D space; the last one is to model the 3D-2D projection implicitly through pure network. .

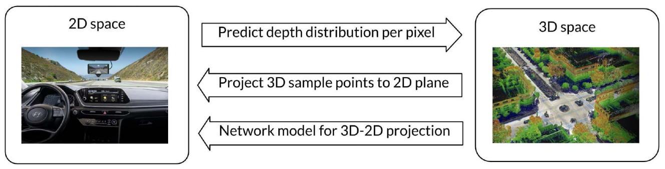

2D space
Predict depth distribution per pixel
Project 3D sample points to 2D plane
Network model for 3D-2D projection
3D space

Fig. 3. Taxonomy of View Transformation. From the **2D-3D** methods, LSS-based approaches [5], [46], [47], [49], [57], [64], [88] predict depth distribution per pixel from 2D feature. From the **3D-2D** methods, homographic matrix based methods [4], [26], [92] presume sparse 3D sample points and project them to 2D plane via camera parameters. **Pure-network-based** methods [94], [95], [96], [97], [98] adopt MLP or transformer to implicitly model the projection from 3D space to 2D plane.

and its temporal version BEVDet4D [64], BEVDepth [49], BEVFusion [5], [88], and others [65], [80], [89]. Note that in stereo setting, depth value/distribution is much more easier to be obtained via the strong prior where the distance between the pair of cameras (namely *baseline* of the system) are supposed to be constant. This can be formulated as:

$$\mathcal{D}(u,v) = f \times \frac{b}{d(u,v)}, \tag{7}$$

where $d(u,v)$ is the horizontal disparity on the pair of images at location (u,v) (usually defined in the left image), f is the focal length of cameras as in Section B of Appendix, available online, $\mathcal{D}(u,v)$ is the depth value at (u,v), and b is the length of the baseline. Stereo methods such as LIGA Stereo [89] and DSGN [65] utilize this strong prior and perform on par with LiDAR-based alternatives on KITTI leaderboard [11].

(2) *3D-2D methods:* The second branch (3D to 2D) can be originated back to thirty years ago, when Inverse Perspective Mapping (IPM) [90] formulated the projection from 3D space to 2D space conditionally hypothesizing that the corresponding points in 3D space lie on a horizontal plane. Such a transformation matrix can be mathematically derived from camera intrinsic and extrinsic parameters [91], and the detail of this process is presented in Section B of Appendix, available online. A series of work apply IPM to transform elements from perspective view to bird's eye view in an either pre-processing or post-processing manner. In context of view transformation, feature projection method from 3D to 2D is first introduced by OFT-Net [43]. OFT-Net forms a uniformly distributed 3D voxel feature grid, populating voxels by aggregating image features from corresponding projection regions. The orthographic BEV feature map is then acquired by summing voxel features vertically. Recently, inspired by Tesla's technical roadmap for the perception system [6], a combination of 3D-2D geometric projection and neural network becomes popular [4], [26], [85], [92]. Note that the cross-attention mechanism in transformer architecture conceptually meets the need of such a geometric projection, as denoted by:

$$\mathcal{F}_{3D}(x,y,z) = CrossAttn(q : P_{xyz}, k\, v : \mathcal{F}^{*}_{2D}(\hat{u},\hat{v})), \tag{8}$$

where q, k, v stand for query, key and value, P_{xyz} is pre-defined anchor point in voxel space, and other notations follow (4) and (5). Some approaches [4], [85] utilize camera parameters to project P_{xyz} to image plane for fast convergence of the model. To obtain robust detection results, BEVFormer [4] exploits the cross-attention mechanism in transformer to enhance the modeling of *3D-2D* view transformation. Others [50], [93] alleviate grid sampler to accelerate this process efficiently for massive production. Nonetheless, these approaches heavily rely on the precision of camera parameters, which are susceptible to fluctuations over extended durations of driving.

(3) *Pure-network-based methods:* Regardless of 2D-3D methods or 3D-2D methods, both techniques introduce inherit inductive bias contained in geometric projection. In contrast, some methods tend to ultilize neural networks for the implicit representation of camera projection relationships. A lot of BEV map segmentation works [55], [56], [94] utilize either multi-layer perceptron or transformer [14] architecture to model the 3D-2D projection implicitly. VPN [94] introduces view relation

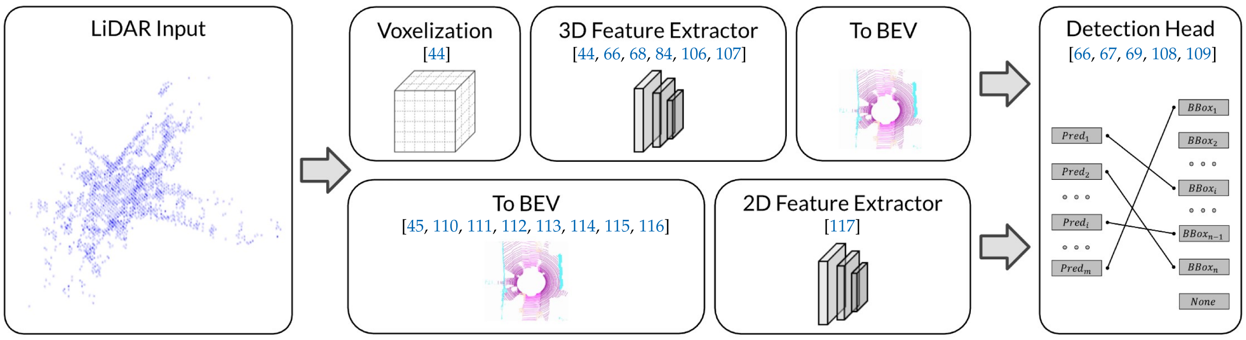


Fig. 4. General pipeline of BEV LiDAR perception. There are mainly two branches to convert point cloud data into BEV representation. The upper branch extracts point cloud features in 3D space, providing more accurate detection results. The lower branch extracts BEV features in 2D space, providing more efficient networks.

module–a Multilayer Perceptron (MLP) for producing map-view features by processing inputs from all views, thus achieves the acquisition of a shared feature representation spanning various perspectives. HDMapNet [55] employs MLP architecture to execute the view transformation of feature maps. BEVSegFormer builds dense BEV queries, and directly predicts their 2D projection points from query feature through MLP, then update the query embedding using deformable attention. CVT [54] incorporate image features with camera-aware position embedding derived from the camera intrinsic and extrinsic parameters, and introduces a cross-view attention module to produce map-view representation. Some methods do not explicitly construct BEV features. PETR [48] integrates 3D positional embedding, derived from camera parameters, into 2D multi-view features. This integration empowers sparse queries to directly interact with 3D position-aware image features through the vanilla cross attention.

3) Discussion on BEV and Perspective Methods: At the very beginning of camera-only 3D perception, the main focus is how to predict 3D object localization from perspective view, *a.k.a. 2D space*. It is because 2D perception is well developed at that stage [1], [2], [99], [100], and how to equip a 2D detector with the ability to perceive 3D scene becomes mainstream methods [62], [81], [82], [101]. Later, some research reached to BEV representation, since under this view, it is easy to tackle the problem where objects with the same size in 3D space have very different size on image plane due to the distance to camera. This series of work [43], [46], [65], [86], [89] either predict depth information or utilize 3D prior assumption to compensate the loss of 3D information in camera input. While recent BEV-based methods [3], [4], [5], [47], [49], [88], [102] has taken the 3D perception world by storm, it is worth noticing that this success has been beneficial from mainly three parts. The first reason is the trending nuScenes dataset [7], which has multi-camera setting and it is very suitable to apply multi-view feature aggregation under BEV. The second reason is that most camera-only BEV perception methods have gained a lot of help from LiDAR-based methods [44], [45], [67], [83], [84], [103], [104] in form of detection head and corresponding loss design. The third reason is that the long-term development of monocular methods [81], [82], [101] has flourished BEV-based methods a good starting point in form of handling feature representation in the perspective view. The core issue is how to reconstruct the lost 3D information from 2D images. To this end, BEV-based methods and perspective methods are two different ways to resolve the same problem, and they are not excluding each other.

B. BEV LiDAR

1) General Pipeline: Fig. 4 depicts the general pipeline for BEV LiDAR perception. The extracted point cloud features are transformed into BEV feature maps. The common detection heads generate 3D prediction results. In terms of the feature extraction part, there are mainly two branches to convert point cloud data into BEV representation. According to the pipeline order, we term these two options as pre-BEV and post-BEV respectively, indicating whether the input of backbone network is from 3D representation or BEV representation.

2) Pre-BEV Feature Extraction: Besides point-based methods processing on the raw point cloud, voxel-based methods voxelize points into discrete grids, which provides a more efficient representation by discretizing the continuous 3D coordinate. Based on the discrete voxel representation, 3D convolution or 3D sparse convolution [117], [118] can be used to extract point cloud features. We use $Y_{j,c'}$ to represent the j-th voxel output Y at output channel c', and $X_{i,c}$ to represent the i-th voxel input X at input channel c. A normal 3D convolutional operation can be described as:

$$Y_{j,c'} = \sum_{i \in P(j)} \sum_{c} W_{k,c,c'} X_{i,c}, \tag{9}$$

where $P(j)$ denotes a function for obtaining the input index i and the filter offset, and $W_{k,c,c'}$ denotes filter weight with kernel offset k. For sparse input $\tilde{X}$ and output $\tilde{Y}$, we can rewrite (9) into 3D sparse convolution:

$$\tilde{Y}_{j,c'} = \sum_{k} \sum_{c} W_{k,c,c'} \tilde{X}_{R_{k,j},k,c}, \tag{10}$$

where $R_{k,j}$ denotes a matrix that specifies the input index i given the kernel offset k and the output index j. Most state-of-the-art methods normally utilize 3D sparse convolution to conduct feature extraction. The 3D voxel features can then be formatted as a 2D tensor in BEV by densifying and compressing the height axis.

VoxelNet [44] stacks multiple voxel feature encoding (VFE) layers to encode point cloud distribution in a voxel as a voxel feature. Given $\mathbf{V} = \{\mathbf{p_i} = [x_i, y_i, z_i, r_i]^T\}_{i=1\ldots n}$ as $n \leq N$ points inside a non-empty voxel, where x_i, y_i, z_i are coordinates in 3D space, r_i is reflectance, N is the maximal number of points, and the centroid of $\mathbf{V}$ (v_x, v_y, v_z) is the local mean of all points, the feature of each point is calculated by:

$$f_i = FCN([x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z]^T). \quad (11)$$

FCN is the composition of a linear layer, a batch normalization, and an activation function. Feature of the voxel is the element-wise max-pooling of all f_i of $\mathbf{V}$. A 3D convolution is applied to further aggregate local voxel features. After merging the dimension of channel and height, the feature maps, which are transformed implicitly into BEV, are processed by a region proposal network (RPN) to generate object proposals. SECOND [84] introduces sparse convolution in processing voxel representation to reduce training and inference speed by a large margin. CenterPoint [67], which is a powerful center-based anchor-free 3D detector, also follows this detection pattern and becomes a baseline method for 3D object detection.

PV-RCNN [66] combines point and voxel branches to learn more discriminative point cloud features. Specifically, high-quality 3D proposals are generated by the voxel branch, and the point branch provides extra information for proposal refinement. SA-SSD [105] designs an auxiliary network, which converts the voxel features in the backbone network back to point-level representations, to explicitly leverage the structure information of the 3D point cloud and ease the loss in downsampling. Voxel R-CNN [107] adopts 3D convolution backbone to extract point cloud feature. A 2D network is then applied on the BEV to provide object proposals, which are refined via extracted features. It achieves comparable performance with point-based methods. Object DGCNN [108] models the task of 3D object detection as message passing on a dynamic graph in BEV. After transforming point cloud into BEV feature maps, predicted query points collect BEV features from key points iteratively. VoTr [106] introduces Local Attention, Dilated Attention, and Fast Voxel Query to enable attention mechanism on numerous voxels for large context information. SST [68] treats extracted voxel features as tokens and then applies Sparse Regional Attention and Region Shif in the non-overlapping region to avoid downsampling for voxel-based networks. AFDetV2 [69] formulates a single-stage anchor-free network by introducing a keypoint auxiliary supervision and multi-task head.

3) Post-BEV Feature Extraction: As voxels in 3D space are sparse and irregular, applying 3D convolution is inefficient. For industrial applications, operators such as 3D convolution may not be supported; suitable and efficient 3D detection networks are desirable.

MV3D [109] is the first method to convert point cloud data into a BEV representation. After discretizing points into the BEV grid, the features of height, intensity, and density are obtained according to points in the grid to represent grid features. As there are many points in a BEV grid, in this processing, the loss of information is considerable. Other works [110], [111], [112], [113], [114], [115] follow the similar pattern to represent point cloud using the statistic in a BEV grid, such as the maximum height and mean of intensity.

PointPillars [45] first introduces the concept of pillar, which is a special type of voxel with unlimited height. It utilizes a simplified version of PointNet [103] to learn a representation of points in pillars. The encoded features can then be processed by standard 2D convolutional networks and detection heads. Though the performance of PointPillars is not as satisfactory as other 3D backbones, it and its variants enjoy high efficiency and thus are suitable for industrial applications.

4) Discussion: The point cloud data are directly processed by the neural network, as does in [119], [120]. The neighborhood relationship among points is calculated in the continuous 3D space. This brings extra time consumption and limits the receptive field of the neural network. Recent works [44], [84] utilize discrete grids to represent point cloud data; convolutional operations are adopted to extract features. However, converting point cloud data into any form of representation inevitably causes the loss of information. State-of-the-art methods in pre-BEV feature extraction utilize voxels with fine-grained size, preserving most of the 3D information from point cloud data and thus beneficial for 3D detection. As a trade-off, it requires high memory consumption and computational cost. Transforming point cloud data directly into BEV representation avoids complex operation in the 3D space. As the height dimension is compressed, a great loss of information becomes inevitable. The most efficient method is to represent the BEV feature map using statistics, and yet it provides inferior results. Pillar-based methods [45] balance performance and cost, and become a popular choice for industrial applications. How to deal with the trade-off between performance and efficiency becomes a vital challenge for LiDAR-based applications.

C. BEV Fusion

1) General Pipeline: Inverse perspective mapping (IPM) [90] is proposed to map pixels onto the BEV plane using the geometric constraint of the intrinsic and extrinsic matrix of cameras. Despite its inaccuracy due to the flat-ground assumption, it provides the possibility that images and point clouds can be unified in BEV. Lift-splat-shoot (LSS) [57] is the first method to predict depth distribution of image features, introducing neural networks to learn the ill-posed camera-to-lidar transformation problem. Other works [4], [121] develop different method to conduct view transformation. Given the view transformation methods from perspective view to BEV, Fig. 5(b) shows the general pipeline for fusing image and point cloud data. Modal-specific feature extractors are used to extract features in perspective view and BEV separately. After transforming to the representations in BEV, feature maps from different sensors are fused. The temporal and ego-motion information can be introduced in the BEV representation as well.

2) LiDAR-Camera Fusion: Concurrently, two works with the same name BEVFusion [5], [88] explore fusion in BEV from different directions. As camera-to-lidar projection [73], [122]

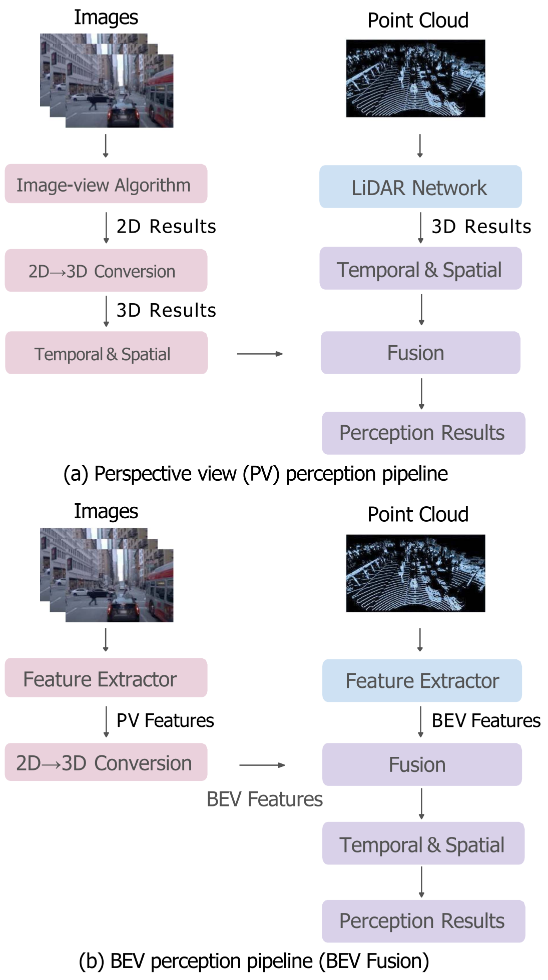


Fig. 5. Two typical pipeline designs for BEV fusion algorithms, applicable to both academia and industry. The main difference lies in 2D to 3D conversion and fusion modules. In the PV perception pipeline (a), results of different algorithm are first transformed into 3D space, then fused using prior or hand-craft rules. The BEV perception pipeline (b) first transforms PV features to BEV, then fuses features to obtain the ultimate predictions, thereby maintaining most original information and avoiding hand-crafted design.

throws away the semantic density of camera features, BEV-Fusion [5] designs an efficient camera-to-BEV transformation method, which efficiently projects camera features into BEV, and then fuses it with lidar BEV features using convolutional layers. BEVFusion [88] regards the BEV fusion as a robustness topic to maintain the stability of the perception system. It encodes camera and lidar features into the same BEV to ensure the independence of camera and lidar streams. This design enables the perception system to maintain stability against sensor failures.

Apart from BEVFusion [5], [88], UVTR [121] represents different input modalities in modal-specific voxel spaces without height compression to avoid the semantic ambiguity and enable further interactions. Image voxel space is constructed by transforming the image feature of each view to the predefined space with depth distribution generated for each image. Point voxel space is constructed using common 3D convolutional networks. Cross-modality interaction is then conducted between two voxel spaces to enhance modal-specific information.

3) Temporal Fusion: Temporal information plays an important role in inferring the motion state of objects and recognizing occlusions. BEV provides a desirable bridge to connect scene representations in different timestamps, as the central location of the BEV feature map is persistent to ego car. MV-FuseNet [123] utilizes both BEV and range view for temporal feature extraction. Other works [53], [63], [64] use ego-motion to align the previous BEV features to the current coordinates, and then fuse the current BEV features to obtain the temporal features. BEVDet4D [64] fuses the previous feature maps with the current frame using a spatial alignment operation followed by a concatenation of multiple feature maps. BEVFormer [4] and UniFormer [124] adopt a soft way to fusion temporal information. The attention module is utilized to fuse temporal information from previous BEV feature maps and previous frames, respectively. Concerning the motion of ego car, locations for the attention module to attend among representations of different timestamps are also corrected by the ego-motion information.

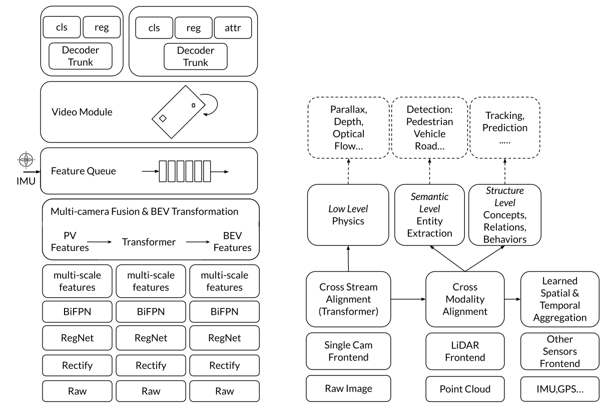
4) Discussion: As images are in perspective coordinate and point clouds are in 3D coordinate, spacial alignment between two modalities becomes a vital problem. Though it is easy to project point cloud data onto image coordinates using geometric projection relationships, the sparse nature of point cloud data makes extracting informative features difficult. Inversely, transforming images in perspective view into 3D space would be an ill-posed problem, due to the lack of depth information in perspective view. Based on prior knowledge, previous work such as IPM [90] and LSS [57] make it possible to transform information in the perspective view into BEV, providing a unified representation for multi-sensor and temporal fusion.

Fusion in the BEV space for lidar and camera data provides satisfactory performance for the 3D detection task. Such a method also maintains the independence of different modalities, which provides the opportunity to build a more robust perception system. For temporal fusion, representations in different timestamps can be directly fused in the BEV space by concerning ego-motion information. Compensation for ego-motion is easy to obtain by monitoring control and motion information, as the BEV coordinate is consistent with the 3D coordinate. With the concern of robustness and consistency, BEV is an ideal representation for multi-sensor and temporal fusion.

D. Industrial Design of BEV Perception

Recent years have witnessed trending popularity for BEV perception in the industry. In this section, we describe the architecture design for BEV perception on a system level.

Fig. 5 depicts two typical paradigms for sensor fusion in the industrial applications. Prior to BEV perception research, most autonomous driving companies construct the perception system based on perspective view inputs. As shown in Fig. 5(a), in the perspective view (PV) pipeline, LiDAR tracks provide 3D results directly, while image-based 3D results are converted from 2D results via geometry prior. Then the predictions from images and LiDAR are fused through hand-crafted approaches. On the contrary, BEV based methods, as illustrated in Fig. 5(b), perform


cls
reg
Decoder Trunk
cls
reg
attr
Decoder Trunk
Video Module
IMU
Feature Queue
Multi-camera Fusion & BEV Transformation
PV Features
Transformer
BEV Features
multi-scale features
BiFPN
RegNet
Rectify
Raw
Parallax, Depth, Optical Flow...
Detection: Pedestrian Vehicle Road...
Tracking, Prediction .....
Low Level Physics
Semantic Level Entity Extraction
Structure Level Concepts, Relations, Behaviors
Cross Stream Alignment (Transformer)
Cross Modality Alignment
Learned Spatial & Temporal Aggregation
Single Cam Frontend
LiDAR Frontend
Other Sensors Frontend
Raw Image
Point Cloud
IMU,GPS...

(a) Video predictor with vision input only [6]

(b) Multi-modality input with various perception tasks [127]

Fig. 6. BEV architecture comparison from industry solutions. These paradigms follow similar workflow as illustrated in Fig. 5(b). There are slight differences in each design. Fig. 6(a) from Tesla [6] takes vision as main input and incorporates video module, while Fig. 6(b) from Horizon [125] embraces multi-modality to tackle multiple perception tasks.

feature-level 2D to 3D transformation and integrate features instead of the direct detection outputs from different modalities, leading to less hand-crafted design and more robustness.

Fig. 6 summarizes various BEV perception architecture proposed by corporations around the globe. The detailed model/input options are described in Section D of Appendix, available online. Note that all the information presented in this survey are collected from public resource; comparison and analysis among different plans are based on facts. The BEV fusion architectures in Fig. 6 follow the pipeline as depicted in Fig. 5(b), consisting of input data, feature extractor, PV to BEV transformation, feature fusion module, temporal & spatial module and prediction head. We elaborate on each module in details below.

1) Input Data: BEV based perception algorithms support different data modalities, including camera, LiDAR, Radar, IMU and GPS. Camera and LiDAR are the main perception sensors for autonomous driving. Some products use camera only as the input sensor, e.g., Tesla [6], PhiGent [126], Mobileye [127]. The others adopt a suite of camera and LiDAR combination, e.g., Horizon [125], HAOMO [128]. Note that IMU and GPS signals are often adopted for sensor fusion plans [6], [125], [128], as do in Tesla and Horizon, etc.

2) Feature Extractor: The feature extractor serves as transforming raw data to appropriate feature representations, and this module often consists of backbone and neck. There are different selections for backbone and neck. For instance, ResNet [116] in HAOMO [128] and RegNet [129] in Tesla [6] can be employed as the image backbone. The neck could be FPN [130] from HAOMO [128], BiFPN [131] from Tesla [6], etc. As for the point cloud input, pillar based option from HAOMO [128] or voxel based choice from Mobileye [127] are ideal candidates for the backbone.

3) PV to BEV Transformation: There are mainly four approaches to perform view transformation in industry: (a) **Fixed IPM**. Based on the flat ground assumption, a fixed transformation can project PV feature to BEV space. Fixed IPM projection handles ground plane well. However, it is sensitive to vehicle jolting and road flatness. (b) **Adaptive IPM** utilizes the extrinsic parameters of SDV, which are obtained by some pose estimation approaches, and projects features to BEV accordingly. Although adaptive IPM is robust to vehicle pose, it still hypothesizes on the flat ground assumption. (c) **Transformer** based BEV transformation employs dense transformer to project PV feature into BEV space. Such data driven transformation-based methods are widely adopted by Tesla, Horizon, HAOMO. (d) **ViDAR** is first proposed in early 2018 by Waymo and Mobileye in parallel at different venues [13], [127], to indicate the practice of using pixel-level depth to project PV feature to BEV space based on camera or vision inputs, resembling the representation form as does in LiDAR. The term ViDAR is equivalent to the concept of pseudo-LiDAR presented in most academic literature. Equipped With ViDAR, one can transform images and subsequently features into point cloud directly. Then point cloud based methods can be applied to get BEV features. We have seen many ViDAR applications [6], [13], [127], [132], [133] recently, e.g., Tesla, Mobileye, Waymo, Toyota, etc. Overall, the options of Transformer and ViDAR are most prevailing in industry.

4) Fusion Module: The alignment among different camera sources has been accomplished in the previous BEV transformation module. In the fusion unit, they step further to aggregate BEV features from camera and LiDAR. By doing so, features from different modalities are ultimately integrated into one unified form.

5) Temporal & Spatial Module: By stacking BEV features temporally and spatially, a feature queue can be constructed. The temporal stack pushes and pops a feature blob every fixed time, while the spatial stack does it every fixed distance. After fusing feature in these stacks into one form, they can obtain a spatial-temporal BEV feature, which is robust to occlusion [6], [128]. The aggregation module can be in form of 3D convolution, RNN or Transformer. Based on the temporal module and vehicle kinematics, one can maintain a large BEV feature map surrounding ego vehicle and update the feature map locally, as does in the spatial RNN module from Tesla [6].

6) Prediction Head: In BEV perception, the multi-head design is widely adopted. Since BEV feature aggregates information from all sensors, all 3D detection results are decoded from BEV feature space. In the meanwhile, PV results are also decoded from the corresponding PV features in some design. The prediction results can be classified into three categories [125]: (a) **Low level results** are related to physics constrains, such as optical flow, depth, etc. (b) **Entity level results** include concepts of objects, i.e., vehicle detection, laneline detection, etc. (c) **Structure level results** represent relationship between objects, including object tracking, motion prediction, etc.

IV. Empirical Evaluation and Recipe

In this section, we summarize the bag of tricks and the most useful practices to achieve top results on various benchmarks. This is based on our contest entries to Waymo Open Challenge

TABLE IV
BEV CAMERA DETECTION TRACK

ID	DeD	FrA	CeP	ConvO	DE	CoP	DA	GR	MS	FL	SL	EMA	SB	2BS	LLW	LS	LE	TTA	Backbone	DS	LET-mAPL	LET-mAPH	L1/mAPH
0	✓																		R101	mini	34.6	46.1	25.5
1	✓			✓															R101	mini	35.9	48.1	25.6
2	✓				✓														R101	mini	36.1	48.1	25.9
3	✓					✓													R101	mini	35.6	46.9	26.0
4	✓						✓								✓				R101	mini	36.2	48.1	25.4
5	✓							✓											R101	mini	35.4	47.2	27.2
6	✓								✓	✓									R101	mini	-	-	26.8
7	✓										✓								R101	mini	-	-	27.3
8	✓											✓							R101	mini	-	-	26.2
9	✓												✓						R101	mini	-	-	25.6
9	✓													✓					R101	mini	-	-	25.5
10	✓														✓				R101	mini	-	-	26.5
11	✓															✓			R101	mini	36.0	46.7	-
12	✓																✓		R101	mini	34.7	44.2	-
13	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	R101	mini	-	-	37.5
14*	✓																		SwinL	mini	40.0	55.6	51.9
15*	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					SwinL	mini	44.7	60.8	55.5
16		✓																	R101	mini	35.9	49.9	45.9
17		✓															✓		R101	mini	36.3	51.1	46.6
18			✓																R101	mini	34.0	47.9	43.5
19	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					SwinL	full	48.4	64.8	60.4
20		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					SwinL	full	47.2	61.2	56.8
21		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓		SwinL	full	47.6	61.4	57.0
22			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					SwinL	full	41.9	54.6	48.2

Ablation studies on val set with improvements over BEVFormer [4], i.e., BEVFormer++. Some results are only reported as L1/mAPH on car category with iou≥0.5. DeD (Deformable DETR head). FrA (FreeAnchor head). CeP (Centerpoint head). ConvO (Conv offsets in Temporal Self Attention (TSA)). DE (Deformable view Encoder). CoP (Corner Pooling). DA (2D Auxiliary loss). GR (Global location regression). MS (Multi-Scale), FL (filp), SL (Smoooth L1 Loss), EMA (Exponential Moving Average), SB (Sync BN), 2BS (2x BEV Scale), LLW(Leanable LossWeight), LS (Label Smoothing), LE (LET-IoU based Assignment) and TTA(Test Time Augmentation). DS (Dataset). The mini dataset contains 1/5 training data. *denotes the model is trained with 24 epochs, otherwise with 12 epochs.

2022, namely BEVFormer++ built on top of BEVFormer [4] for camera-only detection and Voxel-SPVCNN derived from SPVCNN [83] for LiDAR segmentation. These practical experiences can serve as a reference to seamlessly integrate into other BEV perception models and evaluate the efficacy. We present practical experiences related to data augmentation, BEV encoder, loss selection in the following content, and place additional practical experiences regarding detection head design, test-time augmentation, model ensemble and post-processing in Section E of Appendix, available online.

A. Data Augmentation

Data augmentation plays a crucial role in enhancing the robustness and generalization capabilities of perception models. By synthetically expanding the diversity of the training dataset, we can improve the model's ability to handle variations in real-world scenarios. Certain non-affine transformations are challenging to apply for data augmentation for images or BEV space. For instance, methodologies like copy-paste and Mosaic are problematic due to their potential to cause misalignment between image semantics and 3D spatial information.

1) BEV Camera (Camera-Only) Detection: Common data augmentations on images for 2D recognition tasks are applicable for the tasks of camera based BEV perception. In general, we can divide augmentations into static augmentation involving color variation alone and spatial transformation moving pixels around. Augmentations based on color variation are directly applicable. For augmentations involving spatial transformation, apart from ground truth transformed accordingly, calibration in camera parameter is also necessary. Common augmentations adopted in recent camera-only methods are color jitter, flip, resize, rotation, crop, and Grid Mask. In BEVFormer++, color jitter, flip, multi-scale resize and grid mask are employed. The input image is scaled by a factor between 0.5 and 1.2, flipped by a ratio of 0.5; the maximum 30% of total area is randomly masked with square masks. Notably, there are two ways for flipping images in BEV perception, which we refer to as image-level flipping and BEV-level flipping. During image-level flipping, we flip the image and adjust camera intrinsic parameters, while camera extrinsic parameters and GT boxes remain unchanged. This preserves the 3D-to-2D projection relationship. Note that image-level flipping only enhances 2D feature extraction without exerting any influence on the subsequent modules associated with BEV. During BEV-level flipping, we flip the image and rearrange multi-view images symmetrically, such as switching the front-left camera to the front-right. Overlapped area coherence is preserved. Camera intrinsic parameters stay constant, while extrinsic parameters are adjusted, and GT boxes on the BEV plane are flipped. BEV-level flipping boosts the entire BEV perception model. BEV-level flipping is adopted in BEVFormer++. Related ablation study is described in Table IV ID 6 experiment, indicating that data augmentation plays a vital role in improving 3D model's performance. Since BEVFormer [4] employs sequence input, it ensures that the transformation is consistent for each frame of the sequence after input augmentation.

2) LiDAR Segmentaion: Different from the task of detection, heavy data augmentation can be applied in the task of segmentation, including random rotation, scaling, flipping, and point translation. For random rotation, an angle is picked from the range of $[0, 2\pi)$, rotation is applied to every point on the x-y plane. A scale factor is chosen from the range of $[0.9, 1.1]$, and then multiplied on the point cloud coordinate. Random flipping

TABLE V
BEV LiDAR Segmentation Track

ID	Aug	Loss	Arch	TTA	Painting	Temporal	V-SPV	Ensemble	Expert	Post	mIoU
0											67.4
1	✓										67.8
2	✓	✓									68.4
3	✓	✓	✓								69.6
4	✓	✓	✓	✓							71.1
5	✓	✓	✓	✓	✓						71.6
6	✓	✓	✓	✓	✓	✓					72.4
7	✓	✓	✓	✓	✓	✓	✓				73.5
8	✓	✓	✓	✓	✓	✓	✓	✓			74.2
9	✓	✓	✓	✓	✓	✓	✓	✓	✓		74.5
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	75.4

Ablation studies on val set with improvements over SPVCNN [83], i.e., Voxel-SPVCNN. Aug (heavy data augmentation). Arch (adjustments on model architecture). TTA (test-time augmentation). Painting (one-hot painting from image semantic segmentation). Temporal (multi-frame input). V-SPV (Voxel-SPVCNN). Expert (ensemble with more expert models). Post (post-processing techniques including object-level refinement and segmentation with tracking).

is conducted along X axis, Y axis, or both X and Y axes. For random translation, offsets for each axis are sampled separately from a normal distribution with a mean of 0 and a standard deviation of 0.1.

Besides coordinates and reflectance, extra information can be utilized to boost model performance. Painting [73], [122] is a common technique to enhance point cloud data with image information. For unlabeled image data, by projecting point cloud labels onto corresponding images and densifying the sparse annotations, semantic labels on images are obtained from annotated point cloud data. An image model is trained to provide 2D semantic segmentation results. Then, predicted semantic labels are painted as one-hot vectors to point cloud data as additional channels to represent semantic information from images. Besides, temporal information can also be used, as datasets in autonomous driving are usually collected sequentially. Past consecutive frames are concatenated with the current frame. An additional channel is appended to represent the relative time information of different frames. To reduce the number of points, a small voxelization network is applied. Then, voxels treated as points are served as input to our models.

As shown in Table V ID 1, heavy data augmentation brings an improvement of mIoU of 0.4. By introducing information from images and previous frames, gains in model performance are 0.5 and 0.8 mIoU, respectively (Table V ID 5 & 6). It indicates that extra information, especially temporal information, is beneficial for the per-point classification task.

B. BEV Encoder

1) BEV Camera. BEVFormer++: BEVFormer++ has multiple encoder layers, each of which follows the conventional structure of transformers [14], except for three tailored designs, namely BEV queries, spatial cross-attention, and temporal self-attention. Specifically, BEV queries are grid-shaped learnable parameters, which is designed to query features in BEV space from multi-camera views via attention mechanisms. Spatial cross-attention and temporal self-attention are attention layers working with BEV queries, which are used to lookup and aggregate spatial features from multi-camera images as well as temporal features from history BEV feature.

During inference, at timestamp t, we feed multi-camera images to the backbone network (e.g., ResNet-101 [116]), and obtain the features $F_t = \{F_t^i\}_{i=1}^{N_{view}}$ of different camera views, where F_t^i is the feature of the i-th view, N_{view} is the total number of camera views. At the same time, we preserve BEV features B_{t-1} at the prior timestamp $t-1$. In each encoder layer, we first use BEV queries Q to query the temporal information from the prior BEV features B_{t-1} via the temporal self-attention. We then employ BEV queries Q to inquire about the spatial information from multi-camera features F_t via the spatial cross-attention. After the feed-forward network [14], the encoder layer generates the refined BEV features, which is consequently the input of the next encoder layer. After six stacking encoder layers, unified BEV features B_t at current timestamp t are generated. Taking the BEV features B_t as input, the 3D detection head and map segmentation head predict the perception results such as 3D bounding boxes and semantic map.

To improve the feature quality contributing from BEV encoder, three main aspects are to be discussed as follows.

(a) 2D Feature Extractor: Techniques for improving backbone representation quality in 2D perception tasks are most likely to improve presentation quality for BEV tasks as well. For convenience, in the image backbone we adopt feature pyramid that is widely used in most 2D perception tasks. As depicted in Table IV, the structural design of 2D feature extractor, e.g. state-of-the-art image feature extractor [134], global information interaction [135], multi-level feature fusion [130], [136] etc. all contribute to better feature representation for BEV perception. Apart from the structural design, auxiliary tasks supervising backbones are also important for the performance of BEV perception, which will be discussed in Section IV-C1.

(b) View transformation: The transformation takes in image features and reorganizes them into BEV space. Hyper-parameters, including the image feature's sampling range and frequency, as well as BEV resolution are of vital importance for the performance of BEV perception. The sampling range decides how much of the viewing frustum behind an image will be sampled into BEV space. By default this range is equal to the effective range of LiDAR annotation. When efficiency is of higher priority, the upper z-axis part of the viewing frustum can be compromised since it only contains unimportant information such as sky in most cases. The sampling frequency decides the utility of image features. Higher frequency ensures the model to accurately sample corresponding image features for each BEV location with the cost of higher computation. BEV resolution decides the representation granularity of the BEV

feature, where each feature can be accurately traced back to a grid in the world coordinates. High resolution is required for better representation of small scale objects such as traffic lights and pedestrians. Related experiments are depicted in Table IV ID 2&3. In view transformation, feature extraction operations, e.g. convolution block or Transformer block are also present in many BEV perception networks. Adding better feature extraction sub-networks in the BEV space can also improve the BEV perception performance.

(c) Temporal BEV fusion: Given the structure of BEV feature, temporal fusion in BEV space often leverages ego-car pose information to align temporal BEV features. However, other agents' movements are not modeled explicitly in this alignment process and it requires additional learning by the model. As a result, to enhance fusion on features of other moving agents, it is reasonable to increase the perception range of cross-attention as we perform temporal fusion. For instance, we might enlarge the kernel size of attention offset in the deformable attention module or use global attention. Related improvements can be observed in Table IV ID 1.

2) *BEV LiDAR. Voxel-SPVCNN*

Existing 3D perception models are not ideal to recognize small instances due to the coarse voxelization and aggressive downsampling. SPVCNN [83] utilizes Minkowski U-Net [118] in the voxel-based branch. To preserve point cloud resolution, an extra point-based branch without downsampling is used. Features of the point-based and voxel-based branches would be propagated to each other at different stages of the network.

We propose Voxel-SPVCNN by making two effective modifications to the original SPVCNN [83]. Compared to simply performing voxelization on the raw input feature, a lightweight 3-layer MLP is applied to extract point features and then the voxelization process is applied. Besides, the input of the point-based branch is replaced by the voxel-as-point branch. The network structure of this branch is still a MLP; but the input is replaced as voxels. Voxel-SPVCNN is more efficient, as computation on the point-based branch is greatly reduced, especially in the case where the input is multi-scan point cloud. The change of model architecture brings an improvement of 1.1 mIoU (see Table V ID 7).

C. Loss

1) *BEV Camera-Only Detection*

One of the versatile benefits from BEV feature representation is to enable models to be trained with losses proposed in both 2D and 3D object detection. As reported in Table IV ID 16-20, when leveraging different head design, we conclude that the corresponding losses could be transferred with minimum modification, e.g. a tuning in the loss weight.

Apart from the training loss for 3D target, auxiliary loss plays an important role in Camera-only BEV detection. One type of auxiliary loss is to add 2D detection loss on top of 2D feature extractors. Such supervision enhances the localization on 2D image feature, which in turn contributes to 3D representations provided by view transformation in BEV perception. One example of utilizing such auxiliary loss can be observed in Table IV ID 4. Another type of auxiliary loss is depth supervision [49]. When utilizing ground truth depth generated from LiDAR systems, the implicit depth estimation capability of BEV perception can be improved to obtain accurate 3D object localization. Both of these auxiliary tasks can be applied during training to improve performance. As a side note, 2D detection or depth pretrained backbone are commonly adopted as initialization weights [4], [85].

2) *LiDAR Segmentation*

Instead of a conventional cross-entropy loss, Geo loss [137] and Lovász loss [138] are utilized to train all models. To have a better boundary of different classes, Geo loss has a strong response to the voxels with rich details. Lovász loss serves as a differentiable intersection-over-union (IoU) loss to mitigate the class imbalance problem. It improves the model performance by 0.6 mIoU as shown in Table V ID 2.

V. Conlusion

In this survey, we conduct a thorough review on BEV perception in recent years and provide a practical recipe according to our analysis in BEV design pipeline. Grand challenges and future endeavors could be: (a) how to devise a more accurate depth estimator; (b) how to better align feature representations from multiple sensors in a novel fusion mechanism; (c) how to design a parameter-free network such that the algorithm performance is free to pose variation or sensor location, achieving better generalization ablity across various scenarios; and (d) how to incorporate the successful knowledge from foundation models to facilitate BEV perception. More detailed discussion could be found in the Appendix, available online. We hope this work can benefit the community and be an insightful guidebook for future research in 3D perception.

Acknowledgment

This is a joint work between Shanghai Artificial Intelligence Laboratory and SenseTime.

Author Contributions

H. Li, J. Dai and L. Lu lead the project, provide mentorship and allocate resources across task tracks. H. Li, W. Wang, L. Lu drafted the main outline of the project, worked on a preliminary version of the manuscript, supervised key milestones of the project. In Waymo Open Challenge 2022, H. Deng, H. Tian, J. Yang and X. Jia served as team lead. C. Sima, H. Wang, J. Zeng, Z. Li, L. Chen and T. Li are core members during the Challenge, implementing ideas, running experiments, responsible for key outputs of each track. Y. Li is in charge of the whole data processing team. Y. Gao implemented the model ensemble part. For survey writing, J. Zeng wrote the 3D preliminary section. E. Xie wrote and revised the dataset and evaluation part. J. Xie completed the toolbox for bag of tricks. S. Liu and J. Shi oversaw the project from their inception. D. Lin advised on general picture of the research and gave additional comment on the manuscript. Y. Qiao was the primary consultant, eliciting key goals and tracking progress.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2980–2988.

[3] E. Xie et al., "M ^{2} BEV: Multi-camera joint 3D detection and segmentation with unified birds-eye view representation," 2022, *arXiv:2204.05088*.

[4] Z. Li et al., "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," 2022, *arXiv:2203.17270*.

[5] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," 2022, *arXiv:2205.13542*.

[6] Tesla AI Day, 2021. [Online]. Available: https://www.youtube.com/watch?v=j0z4FweCy4M

[7] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11618–11628.

[8] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2443–2451.

[9] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2771–2780.

[10] T. Wang, J. Pang, and D. Lin, "Monocular 3D object detection with depth from motion," 2022, *arXiv:2207.12988*.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[12] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, J. Vanschoren and S. Yeung, Eds. 2021, vol. 1. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/4734ba6f3de83d861c3176a6273cac6d-Paper-round2.pdf

[13] "Drago anguelov – Machine learning for autonomous driving at scale," 2020. [Online]. Available: https://youtu.be/BV4EXwlb3yo

[14] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[15] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv: 2010.11929*.

[16] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[17] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15979–15988.

[18] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[19] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[20] W. Liang, P. Xu, L. Guo, H. Bai, Y. Zhou, and F. Chen, "A survey of 3D object detection," *Multimedia Tools Appl.*, vol. 80, no. 19, pp. 29 617–29 641, 2021.

[21] R. Qian, X. Lai, and X. Li, "3D object detection for autonomous driving: A survey," *Pattern Recognit.*, vol. 130, 2022, Art. no. 108796.

[22] Y. Ma et al., "Vision-centric BEV perception: A survey," 2022, *arXiv:2208.02797*.

[23] J. Mao, S. Shi, X. Wang, and H. Li, "3D object detection for autonomous driving: A review and new outlooks," 2022, *arXiv:2206.09474*.

[24] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8740–8749.

[25] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[26] L. Chen et al., "PersFormer: 3D lane detection via perspective transformer and the openlane benchmark," 2022, *arXiv:2203.11089*.

[27] F. Yan et al., "ONCE-3DLanes: Building monocular 3D lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17 143–17 152.

[28] J. Houston et al., "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot. Learn.*, 2021, pp. 409–418.

[29] Q.-H. Pham et al., "A* 3D dataset: Towards autonomous driving in challenging environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2267–2273.

[30] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9552–9557.

[31] J. Behley et al., "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9296–9306.

[32] J. Geyer et al., "A2d2: Audi autonomous driving dataset," 2020, *arXiv: 2004.06320*.

[33] N. Gählert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler, "Cityscapes 3D: Dataset and benchmark for 9 DoF vehicle detection," 2020, *arXiv: 2006.07864*.

[34] P. Xiao et al., "Pandaset: Advanced sensor suite dataset for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 3095–3101.

[35] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," 2021, *arXiv:2109.13410*.

[36] Z. Wang et al., "Cirrus: A long-range Bi-pattern LiDAR dataset," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5744–5750.

[37] J. Mao et al., "One million scenes for autonomous driving: Once dataset," 2021, *arXiv:2106.11037*.

[38] X. Weng et al., "All-in-one drive: A large-scale comprehensive perception dataset with high-density long-range point clouds," 2021.

[39] T. Wang, W. Ji, S. Chen, G. Chongjian, E. Xie, and P. Luo, "Deep-Accident: A large-scale accident dataset for multi-vehicle autonomous driving," 2022.

[40] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," 2017, *arXiv: 1711.03938*.

[41] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[42] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2432–2443.

[43] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," in *Proc. Brit. Mach. Vis. Conf.*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53755805

[44] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[45] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12689–12697.

[46] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8551–8560.

[47] J. Huang, G. Huang, Z. Zhu, and D. Du, "BEVDet: High-performance multi-camera 3D object detection in bird-eye-view," 2021, *arXiv:2112.11790*.

[48] Y. Liu, T. Wang, X. Zhang, and J. Sun, "PETR: Position embedding transformation for multi-view 3D object detection," 2022, *arXiv:2203.05625*.

[49] Y. Li et al., "BEVDepth: Acquisition of reliable depth for multi-view 3D object detection," 2022, *arXiv:2206.10092*.

[50] D. Rukhovich, A. Vorontsova, and A. Konushin, "ImVoxelNet: Image to voxels projection for monocular and multi-view general-purpose 3D object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1265–1274.

[51] Y. Jiang et al., "PolarFormer: Multi-camera 3D object detection with polar transformers," 2022, *arXiv:2206.15398*.

[52] L. Reiher, B. Lampe, and L. Eckstein, "A sim2Real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–7.

[53] A. Hu et al., "FIERY: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 15253–15262.

[54] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13750–13759.

[55] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HDMapNet: An online HD map construction and evaluation framework," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 4628–4634.

[56] A. Saha, O. Mendez, C. Russell, and R. Bowden, "Translating images into maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 9200–9206.

[57] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 194–210.

[58] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," 2022, *arXiv:2207.07601*.

[59] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3D-LaneNet: End-to-end 3D multiple lane detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2921–2930.

[60] Y. B. Can, A. Liniger, D. P. Paudel, and L. Van Gool, "Structured bird's-eye-view traffic scene understanding from onboard images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 15641–15650.

[61] W.-C. Hung, H. Kretzschmar, V. Casser, J.-J. Hwang, and D. Anguelov, "Let-3D-ap: Longitudinal error tolerant 3d average precision for camera-only 3D detection," 2022, *arXiv:2206.07705*.

[62] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," 2022, *arXiv:2107.14160*.

[63] Y. Zhang et al., "BEVerse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," 2022, *arXiv:2205.09743*.

[64] J. Huang and G. Huang, "BEVDet4D: Exploit temporal cues in multi-camera 3D object detection," 2022, *arXiv:2203.17054*.

[65] C. Yilun, S. Liu, X. Shen, and J. Jia, "DSGN: Deep stereo geometry network for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12533–12542.

[66] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3d object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 529–10 538.

[67] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11779–11788.

[68] L. Fan et al., "Embracing single stride 3D object detector with sparse transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8448–8458.

[69] Y. Hu et al., "AFDetV2: Rethinking the necessity of the second stage for object detection from point clouds," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, pp. 969–979, 2022.

[70] S. Shi et al., "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection," 2021, *arXiv:2102.00463*.

[71] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, "Pyramid R-CNN: Towards better performance and adaptability for 3D object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2703–2712,.

[72] 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," 2022, *arXiv:2205.05979*.

[73] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4603–4611.

[74] T. Yin, X. Zhou, and P. Krähenbühl, "Multimodal virtual point 3d detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 16494–16507.

[75] Z. Chen et al., "AutoAlign: Pixel-instance feature aggregation for multi-modal 3D object detection," 2022, *arXiv:2201.06493*.

[76] X. Bai et al., "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1080–1089.

[77] Y. Li et al., "DeepFusion: LiDAR-camera deep fusion for multi-modal 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17161–17170.

[78] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "AutoAlignV2: Deformable feature aggregation for dynamic multi-modal 3D object detection," 2022, *arXiv:2207.10316*.

[79] K. He, R. Girshick, and P. Dollár, "Rethinking imageNet pre-training," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4917–4926.

[80] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-LiDAR needed for monocular 3D object detection?," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3122–3132.

[81] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3D object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 913–922.

[82] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single-stage monocular 3D object detection via keypoint estimation," 2020, *arXiv: 2002.10111*.

[83] H. Tang et al., "Searching efficient 3D architectures with sparse point-voxel convolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[84] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.

[85] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3D: 3D object detection from multi-view images via 3D-to-2D queries," 2022, *arXiv:2110.06922*.

[86] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8437–8445.

[87] Y. You et al., "Pseudo-lidar++: Accurate depth for 3D object detection in autonomous driving," 2019, *arXiv: 1906.06310*.

[88] T. Liang et al., "BEVFusion: A simple and robust LiDAR-camera fusion framework," 2022, *arXiv:2205.13790*.

[89] X. Guo, S. Shi, X. Wang, and H. Li, "LIGA-Stereo: Learning LiDAR geometry aware representations for stereo-based 3D detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3133–3143.

[90] H. A. Mallot, H. H. Bülthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, no. 3, pp. 177–185, 1991.

[91] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, vol. 30, no. 9/10, pp. 1333–1341, 2001.

[92] S. Gong et al., "GitNet: Geometric prior-based transformation for birds-eye-view segmentation," 2022, *arXiv:2204.07733*.

[93] T. Wang, Q. Lian, C. Zhu, X. Zhu, and W. Zhang, "MV-FCOS3D++: Multi-View camera-only 4D object detection with pretrained monocular backbones," 2022, *arXiv:2207.12716*.

[94] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4867–4873, Jul. 2020.

[95] N. Hendy et al., "Fishing net: Future inference of semantic heatmaps in grids," 2020, *arXiv: 2006.09917*.

[96] K. Chitta, A. Prakash, and A. Geiger, "NEAT: Neural attention fields for end-to-end autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 15773–15783.

[97] W. Yang et al., "Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15531–15540.

[98] N. Gosala and A. Valada, "Bird's-eye-view panoptic segmentation using monocular frontal view images," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 1968–1975, Apr. 2022.

[99] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.

[100] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[101] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2345–2353.

[102] M. H. Ng, K. Radia, J. Chen, D. Wang, I. Gog, and J. E. Gonzalez, "BEV-Seg: Bird's eye view semantic segmentation using geometry and semantic point cloud," 2020, *arXiv: 2006.11436*.

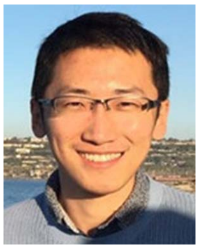
[103] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[104] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[105] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11870–11879.

[106] J. Mao et al., "Voxel transformer for 3D object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3144–3153.
[107] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 2, pp. 1201–1209, 2021.
[108] Y. Wang and J. M. Solomon, "Object DGCNN: 3D object detection using dynamic graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 20745–20758.
[109] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6526–6534.
[110] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 76520–7660.
[111] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," 2018, *arXiv: 2012.11704*.
[112] S. Kahl, M. C. Wood, M. Eibl, and H. Klinck, "BirdNET: A deep learning solution for avian diversity monitoring," *Ecological Inform.*, vol. 61, 2021, Art. no. 101236.
[113] Y. Zeng et al., "RT3D: Real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving," *IEEE Trans. Robot. Autom.*, vol. 3, no. 4, pp. 3434–3440, Oct. 2018.
[114] W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. El Sallab, "Yolo3D: End-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.
[115] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.
[116] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[117] B. Graham, "Spatially-sparse convolutional neural networks," 2014, *arXiv:1409.6070*.
[118] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convNets: Minkowski convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3070–3079.
[119] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9276–9285.
[120] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with pointformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7459–7468.
[121] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3D object detection," 2022, *arXiv:2206.00630*.
[122] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-modal augmentation for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11789–11798.
[123] A. Laddha, S. Gautam, S. Palombo, S. Pandey, and C. Vallespi-Gonzalez, "MVFuseNet: Improving end-to-end object detection and motion forecasting through multi-view fusion of LiDAR data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 2859–2868.
[124] Z. Qin, J. Chen, C. Chen, X. Chen, and X. Li, "UniFormer: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view," 2022, *arXiv:2207.08536*.
[125] Horizon Algorithms, 2022. [Online]. Available: https://en.horizon.ai/products/applications-algorithms/
[126] PhiGent: Technical Roadmap, 2022. [Online]. Available: https://43.132.128.84/coreTechnology
[127] CES 2020 by Mobileye, 2020. [Online]. Available: https://youtu.be/HPWGFzqd7pI
[128] HAOMO AI DAY, 2022. [Online]. Available: https://www.bilibili.com/video/BV1Wr4y1H7W7
[129] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10425–10433.
[130] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.
[131] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10778–10787.
[132] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6612–6619.
[133] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8976–8985.
[134] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.
[135] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
[136] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," *CoRR*, vol. abs/2010.04159, 2020. [Online]. Available: https://arxiv.org/abs/2010.04159
[137] J. Li et al., "Depth based semantic scene completion with position importance aware loss," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 219–226, Jan. 2020.
[138] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4413–4421.
[139] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2147–2156.
[140] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5632–5640.
[141] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D object reconstruction via render-and-compare," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3559–3568.
[142] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv: 1904.07850*.
[143] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9286–9295.
[144] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11859–11868.
[145] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2064–2073.
[146] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.
[147] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021.
[148] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3DNet: 3D object detection using hybrid geometric primitives," in *Proc. Comput. Vis.–ECCV 16th Eur. Conf.*, Glasgow, UK, Aug. 23–28, 2020, pp. 311–329.
[149] B. Cheng, L. Sheng, S. Shi, M. Yang, and D. Xu, "Back-tracing representative points for voting-based 3D object detection in point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8959–8968.
[150] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3D object detection via transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2929–2938.
[151] H. Wang et al., "RBGNet: Ray-based grouping for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1100–1109.
[152] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11037–11045.
[153] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, pp. 87–102, 2018.
[154] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[155] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 828–838.

[156] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6410–6419.
[157] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, “Point transformer,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16239–16248.
[158] Q. Hu et al., “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11105–11114.
[159] Y. Zhang et al., “PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9598–9607.
[160] X. Zhu et al., “Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9934–9943.
[161] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, “(AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12542–12551.
[162] M. Gerdzhev, R. Razani, E. Taghavi, and L. Bingbing, “TORNADO-Net: Multiview total variation semantic segmentation with diamond inception module,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 9543–9549.
[163] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, “AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation,” 2020, *arXiv: 2012.04934*.
[164] M. Ye, S. Xu, T. Cao, and Q. Chen, “DRINet: A dual-representation iterative learning network for point cloud segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 7427–7436.
[165] M. Ye, R. Wan, S. Xu, T. Cao, and Q. Chen, “Drinet++: Efficient voxel-as-point point cloud segmentation,” 2021, *arXiv:2111.08318*.
[166] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, “RPVNet: A deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16004–16013.
[167] K. Genova et al., “Learning 3D semantic segmentation with only 2D image supervision,” in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 361–372.
[168] X. Yan et al., “2DPASS: 2D priors assisted semantic segmentation on LiDAR point clouds,” 2022, *arXiv:2207.04397*.
[169] V. A. Sindagi, Y. Zhou, and O. Tuzel, “MVX-Net: Multimodal voxelnet for 3D object detection,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 7276–7282.
[170] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, “Multi-task multi-sensor fusion for 3D object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7337–7345.
[171] M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 641–656.
[172] R. Nabati and H. Qi, “CenterFusion: Center-based radar and camera fusion for 3D object detection,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1526–1535.
[173] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, “FUTR3D: A unified sensor fusion framework for 3D detection,” 2022, *arXiv:2203.10642*.
[174] Z. Yang, J. Chen, Z. Miao, W. Li, X. Zhu, and L. Zhang, “DeepInteraction: 3D object detection via modality interaction,” 2020, *arXiv:2208.11112*.
[175] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum pointnets for 3D object detection from RGB-D data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[176] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3D proposal generation and object detection from view aggregation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.
[177] S. Pang, D. Morris, and H. Radha, “CLOCs: Camera-LiDAR object candidates fusion for 3D object detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10386–10393.
[178] J. Philion, A. Kar, and S. Fidler, “Learning to evaluate perception models using planner-centric metrics,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14052–14061.
[179] ViDAR – Forward Visual Perception for Advanced Self-Driving, 2022. [Online]. Available: https://43.132.128.84/productions/visualRadar
[180] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
[181] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, “Freeanchor: Learning to match anchors for visual object detection,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 147–155.
[182] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, “DN-DETR: Accelerate DETR training by introducing query deNoising,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13609–13617.
[183] D. Meng et al., “Conditional DETR for fast training convergence,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3631–3640.
[184] R. Solovyev, W. Wang, and T. Gabruseva, “Weighted boxes fusion: Ensembling boxes from different object detection models,” *Image Vis. Comput.*, vol. 107, pp. 104–117, 2021.
[185] Y. Liu et al., “1st place solutions for OpenImage2019–object detection and instance segmentation,” 2020, *arXiv: 2003.07557*.
[186] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: Dynamic and fast instance segmentation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 17721–17732.
[187] “Neural network intelligence,” [Online]. Available: https://github.com/microsoft/nni
[188] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, “Monocular 3D object detection: An extrinsic parameter free approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7552–7562.
[189] R. Bommasani et al., “On the opportunities and risks of foundation models,” 2021, *arXiv:2108.07258*.
[190] T. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.
[191] P. Wang et al., “OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 23318–23340.
[192] J. Zhu et al., “Uni-Perceiver-MoE: Learning sparse generalist models with conditional moes,” 2022, *arXiv:2206.04674*.
[193] S. Reed et al., “A generalist agent,” 2022, *arXiv:2205.06175*.



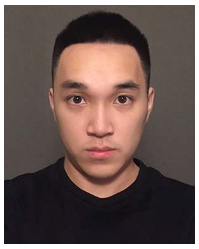
Hongyang Li (Senior Member, IEEE) received the PhD degree from the Chinese University of Hong Kong, in 2019. He is currently a research scientist with OpenDriveLab, Shanghai AI Lab. His expertise focuses on perception and cognition, end-to-end autonomous driving and foundation model. He is also affiliated with Shanghai Jiao Tong University. He serves as area chair for top-tiered conferences multiple times, including CVPR, NeurIPS. He won as PI the CVPR 2023 Best Paper Award, and proposed BEVFormer.

Chonghao Sima received the BE degree in computer science from the Huazhong University of Science and Technology, in 2019. He is currently working toward the PhD degree with the University of Hong Kong. He is currently a researcher with OpenDriveLab, Shanghai AI Lab. His research interests include autonomous driving, computer vision, and foundation model. He was an outstanding reviewer with CVPR 2023. He is the core author of a few popular work, including BEVFormer, OccNet, etc. He won the First Place at Waymo Challenge 2022.

Jifeng Dai received the PhD degree from the Department of Automation, Tsinghua University, in 2014, under the supervision of Professor Jie Zhou. He is an associate professor with the Department of Electronic Engineering, Tsinghua University. His current research focus is on deep learning for high-level vision. Prior to that, he was an executive research director with SenseTime Research, headed by Professor Xiaogang Wang, between 2019 and 2022. He was a principle research manager in Visual Computing Group, Microsoft Research Asia (MSRA) between 2014 and 2019, headed by Dr. Jian Sun.

Wenhai Wang received the PhD degree from Nanjing University. He is a postdoctoral fellow with the Chinese University of Hong Kong. He was a research scientist with the Shanghai AI Laboratory. His main research interests include foundation models, object detection/segmentation, autonomous driving perception, and optical character recognition.

Lewei Lu received the MS degree in computer science from SCUT-MSRA joint-training program, in 2017. He is currently a researcher with SenseTime. He was a researcher with Microsoft Research Asia between 2017 and 2019. His research interests include high-level vision and multi-modal learning and foundation model.


Huijie Wang received the BSc degree in computer science from the Karlsruhe Institute of Technology, in 2019, and the MSc degree in computer science from the Technical University of Munich, in 2022. He is currently a researcher with OpenDriveLab, Shanghai AI Lab. His research interests include autonomous driving and computer vision.

Jia Zeng received the BE degree from Central South University, in 2017, and the PhD degree from Shanghai Jiao Tong University, in 2023. He is currently a researcher with OpenDriveLab, Shanghai AI Laboratory. His research interests include computer vision and autonomous driving.

Zhiqi Li received the BE degree in computer science from Nanjing University, in 2020. He is currently working toward the PhD degree with Nanjing University. He is also a research intern with Shanghai AI lab. His research interests include autonomous driving perception and computer version. He won Frist Place with Waymo Open Dataset Challenge 2022 and the Best Champion at the Occupancy Prediction Task of Autonomous Driving Challenge at CVPR 2023.

Jiazhi Yang received the BE degree in computer science from Sichuan University, in 2022. He is currently a researcher with OpenDriveLab, Shanghai AI Lab. His research interests include autonomous driving and visual intelligence. He is the first author of UniAD that won Best Paper at CVPR 2023. He won the First Place at Waymo Challenge 2022.

Hanming Deng received the MS degree in computer science from Shanghai Jiao Tong University, in 2020. He is currently a researcher with Sensetime. His research interests include focus on perception and planning in autonomous driving. He won the First Place with Waymo Challenge 2022.

Hao Tian received the BE degree in electronic information from the Huazhong University of Science and Technology, in 2018, and the MS degree in mathmatic from the University of Heidelberg in 2022. He is currently a researcher with Sensetime. His research interests include autonomous driving, multi modal foundation model.

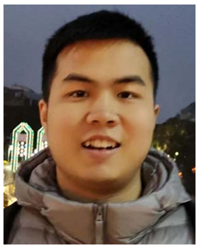
Enze Xie received the BS degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016, and the MS degree from Tongji University, Shanghai, China, in 2019, and the PhD degree from the Department of Computer Science, University of Hong Kong, in 2022. His main research interests include object detection in 2-D and 3-D, and transformers.

Jiangwei Xie received the MS degree in computer science from ShanghaiTech University, in 2022. He is a researcher with the Fundamental Vision Group, Sensetime. His current research interest includes autonomous driving.

Li Chen received the BE degree in mechanical engineering from Shanghai Jiao Tong University, in 2019, and the MS degree in robotics from the University of Michigan, Ann Arbor, USA, in 2020. He is currently a researcher with OpenDriveLab, Shanghai AI Lab. His research interests include autonomous driving and computer vision. He is a recipient with WAIC Yunfan Award, Outstanding Reviewer with CVPR 2023 and the first author of UniAD that won Best Paper at CVPR 2023.

Tianyu Li received the master's degree from Beihang University. He is currently working toward the PhD degree with Fudan University. Additionally, he serves as an intern with OpenDriveLab, Shanghai AI Laboratory. His primary research interests include revolve around autonomous driving and object detection.

Yang Li received the master's degree from Donghua University, and is currently a researcher with OpenDriveLab, Shanghai AI Lab. His research interests include data engine and computer vision.



Yulu Gao received the bachelor's degree from Beihang University. He is currently working toward the PhD degree in computer science and engineering with Beihang University. His research interests include object detection and visual tracking.

Xiaosong Jia received the BE degree in IEEE honor class from Shanghai Jiao Tong University. He is currently working toward the PhD degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai. His research interests include autonomous driving and machine learning, with (co-) first-authored papers published in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, CVPR, ICCV, NeurIPS, RAL, etc.

Si Liu received the PhD degree from the Institute of Automation, Chinese Academy of Sciences. She is a professor with Beihang University. She has been research assistant and postdoc with the National University of Singapore. She was a visiting scholar with Microsoft Research Asia. She has published more than 40 cuttingedge papers on image editing and segmentation on *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, CVPR, ECCV and ICCV. She was the recipient of Best Paper of ACM MM 2013 and 2021, Best demo award of ACM MM 2012. She servers as an area chair of ICCV 2019, CVPR 2020 and ECCV 2020, SPC of IJCAI 2019 and AAAI 2019.

Jianping Shi (Member, IEEE) received the BEng degree from Zhejiang University, in 2011, and the PhD degree from the Computer Science and Engineering Department, Chinese University of Hong Kong, in 2015, under the supervision of Prof. Jiaya Jia. She is currently an executive research director with SenseTime. Her team works on developing algorithms for autonomous driving, scene understanding, and remote sensing. She has served regularly on the organization committees for numerous conferences, such as the Area Chair of CVPR and ICCV.

Dahua Lin received the BEng degree from the University of Science and Technology of China, Hefei, China, in 2004, the MPhil degree from the Chinese University of Hong Kong, Hong Kong, in 2006, and the PhD degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2012. From 2012 to 2014, he was a research assistant professor with Toyota Technological Institute at Chicago, Chicago, IL, USA. He is currently an associate professor with the Department of Information Engineering, Chinese University of Hong Kong (CUHK), and the leading scientist with Shanghai AI Laboratory.

Yu Qiao (Senior Member, IEEE) is a professor with Shanghai AI Laboratory and the Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Science. His research interests include computer vision, deep learning, and bioinformation. He has published more than 300 papers in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Transactions on Image Processing*, CVPR, ICCV etc. His H-index is 72, with 42,700 citations in Google scholar. He is a recipient of the distinguished paper award in AAAI 2021. His group achieved the first runner-up at the ImageNet Large Scale Visual Recognition Challenge 2015in scene recognition, and the winner at the ActivityNet Large Scale Activity Recognition Challenge 2016 in video classification.