Journal of
Sensor and Actuator Networks

MDPI

Article

Online High-Definition Map Construction for Autonomous Vehicles: A Comprehensive Survey

Hongyu Lyu *, Julie Stephany Berrio Perez, Yaoqi Huang, Kunming Li, Mao Shan and Stewart Worrall *

Australian Centre for Robotics, The University of Sydney, Camperdown, NSW 2006, Australia; stephany.berrioperez@sydney.edu.au (J.S.B.); yoki.huang@sydney.edu.au (Y.H.); kunming.li@sydney.edu.au (K.L.); mao.shan@sydney.edu.au (M.S.)
* Correspondence: h.lyu@acfr.usyd.edu.au (H.L.); s.worrall@acfr.usyd.edu.au (S.W.)

Abstract: High-definition (HD) maps aim to provide detailed road information with centimeter-level accuracy, essential for enabling precise navigation and safe operation of autonomous vehicles (AVs). Traditional offline construction methods involve several complex steps, such as data collection, point cloud generation, and feature extraction, but these methods are resource-intensive and struggle to keep pace with the rapidly changing road environments. In contrast, online HD map construction leverages onboard sensor data to dynamically generate local HD maps, offering a bird's-eye view (BEV) representation of the surrounding road environment. This approach has the potential to improve adaptability to spatial and temporal changes in road conditions while enhancing cost-efficiency by reducing the dependency on frequent map updates and expensive survey fleets. This survey provides a comprehensive analysis of online HD map construction, including the task background, high-level motivations, research methodology, key advancements, existing challenges, and future trends. We systematically review the latest advancements in three key sub-tasks: map segmentation, map element detection, and lane graph construction, aiming to bridge gaps in the current literature. We also discuss existing challenges and future trends, covering standardized map representation design, multitask learning, and multi-modality fusion, while offering suggestions for potential improvements.

Keywords: high-definition map; online high-definition map construction; autonomous vehicle; autonomous driving

Academic Editors: Michel Kulhandjian and Hovannes Kulhandjian

Received: 20 November 2024
Revised: 23 January 2025
Accepted: 26 January 2025
Published: 2 February 2025

Citation: Lyu, H.; Berrio Perez, J.S.; Huang, Y.; Li, K.; Shan, M.; Worrall, S. Online High-Definition Map Construction for Autonomous Vehicles: A Comprehensive Survey. *J. Sens. Actuator Netw.* **2025**, *14*, 15. https://doi.org/10.3390/jsan14010015

1. Introduction

High-definition (HD) maps aim to provide highly accurate digital representations of the road environment with centimeter-level precision and multi-dimensional data [1,2]. These maps typically consist of several key layers [3,4]: (1) base map layer, which offers a 3D representation of the environment created from sensor data; (2) geometric map layer, which defines lane-level geometric features such as the layout of roads, lanes, sidewalks, and traffic elements; (3) semantic map layer, which offers semantic context, including road features such as traffic signs, signals, and pedestrian crossings; (4) road connectivity layer, which describes topological relationships between road features, supporting lane connectivity and intersection layouts; (5) prior map layer, which uses historical data to predict road participant behavior and dynamic traffic scenarios; and (6) real-time data layer, which provides live updates, including traffic conditions, weather, and road closures. HD maps serve as long-range sensors for AVs and overcome the limitations of on-board sensors, such as visual occlusion and time constraints. They enhance AVs' ability to perceive and

understand complex traffic scenes, supporting tasks such as localization, path planning, decision-making, and control [5,6]. HD maps can be global, offering broad coverage, or local, providing detailed regional information.

Specifically, traditional global HD maps are built offline, involving three main steps: data collection, point cloud map generation, and feature extraction [7–11]. During data collection, survey vehicles equipped with advanced sensor systems—such as light detection and ranging systems (LiDARs), cameras, radars, global navigation satellite system (GNSS) receivers, and IMUs—are deployed to acquire multimodal environmental data with high precision. Then, the multimodal data are processed through algorithms such as simultaneous localization and mapping (SLAM) [12,13], point cloud registration [14,15], and sensor fusion [16,17] techniques to construct a detailed 3D point cloud representation of the environment. Finally, relevant features such as road networks, lane markings, traffic signs, and traffic lights are extracted from the point cloud map using manual methods or machine learning techniques [120].

While the traditional offline HD map construction method can integrate data from multiple sources and support complex computation and analysis, it also faces some significant drawbacks. Firstly, the process is very costly. The data collection phase requires survey vehicles equipped with advanced sensors, which are expensive to purchase and maintain. Additionally, the feature extraction phase incurs further costs due to the need for personnel for manual labeling or high-performance computing equipment for running machine learning algorithms [7]. Secondly, it is challenging to maintain the accuracy and relevance of HD maps in dynamic environments. Environmental changes, such as new road construction, altered traffic signs, or unexpected accidents, can lead to discrepancies between the map and current conditions. As a result, static maps quickly become outdated, necessitating frequent and costly updates [3,18].

In this context, researchers have proposed algorithms [19–22] that leverage large-scale data and learning-based methods to construct HD maps online, attracting increasing attention. Concisely, we define the corresponding task as **Online HD Map Construction,** which takes raw data from vehicle-mounted sensors as input and generates local HD maps as output. The raw data consist of multimodal sensor outputs, including camera images, LiDAR and radar point clouds, and IMU and GNSS data. The resulting HD map is a bird's-eye view (BEV) representation of the surrounding road environment, primarily in rasterized or vectorized formats. This map includes information about static traffic elements, such as lane dividers, road boundaries, pedestrian crossings, traffic lights, and traffic signs.

The online method has the potential to offer two advantages: improved generalization capability and increased cost efficiency. For generalization capability, it adapts well, both spatially and temporally. Spatially, it can extend knowledge from annotated maps to partially annotated or unannotated areas, predicting road features and structures in unseen regions. Temporally, it adapts to dynamic traffic environments by processing real-time sensor data and updating maps promptly to reflect current conditions [22]. Regarding cost efficiency, the online method can reduce resource demands during map creation and maintenance. During map creation, it streamlines data collection by prioritizing areas with complex road layouts while reducing effort in regions with repetitive or sparse road information, such as highways or remote regions. During map maintenance, it reduces the need for frequent updates by generating maps that reflect real-time conditions, thereby reducing the dependence on costly survey fleets and labor-intensive data annotation [23].

1.1. Comparison to Related Surveys

Although online HD map construction is an emerging and promising research area, comprehensive reviews on this topic are still limited. Recent reviews on BEV perception [24,25] have examined key perception tasks for AVs, such as 3D object detection, map segmentation, and sensor fusion, but have not sufficiently addressed the task of online HD map construction. Further, recent HD map reviews [1,3,7,26] have focused on map structure, functionality, offline construction, and maintenance methods, whereas the techniques of online HD map construction are overlooked in these surveys. Moreover, while some surveys [23,27] have examined online methods for HD map construction, their emphasis on traditional offline techniques might potentially limit the applicability of their findings, given the rapid advancements in this dynamic field. To fill these gaps, we aim to comprehensively review the latest progress in online HD map construction and provide a thorough analysis of the latest achievements, existing challenges, and future trends.

1.2. Contributions

In summary, this study makes three key contributions:

1. We provide a comprehensive analysis of online HD map construction for the first time, covering task background, high-level motivations, research methodology, key advancements, existing challenges, and future trends. We advocate that dynamically generating local HD maps using real-time vehicle sensor data can improve mapping algorithms' generalization and cost efficiency.
2. We systematically review the latest advancements in online HD map construction, focusing on three key sub-tasks: map segmentation, map element detection, and lane graph construction. For each sub-task, we describe method classifications and criteria, discussing the strengths and weaknesses of each approach.
3. We discuss existing challenges and future trends in online HD map construction. We focus on standardized map representation design, multitask learning, and multimodality fusion while suggesting potential improvements.

1.3. Structure

Figure 1 presents the structure of our survey. Section 1 introduces online HD map construction, highlighting its significance and research motivations. Section 2 outlines the research methodology, covering the search strategy, selection criteria, review process, and survey results. Section 3 presents background information, including task definitions, commonly used datasets, and evaluation metrics. Section 4 discusses the latest advancements categorized by the output map format—map segmentation, map element detection, and lane graph construction—analyzing various methods and their strengths and weaknesses. Section 5 addresses current challenges and identifies potential future research directions. Section 6 concludes this paper.

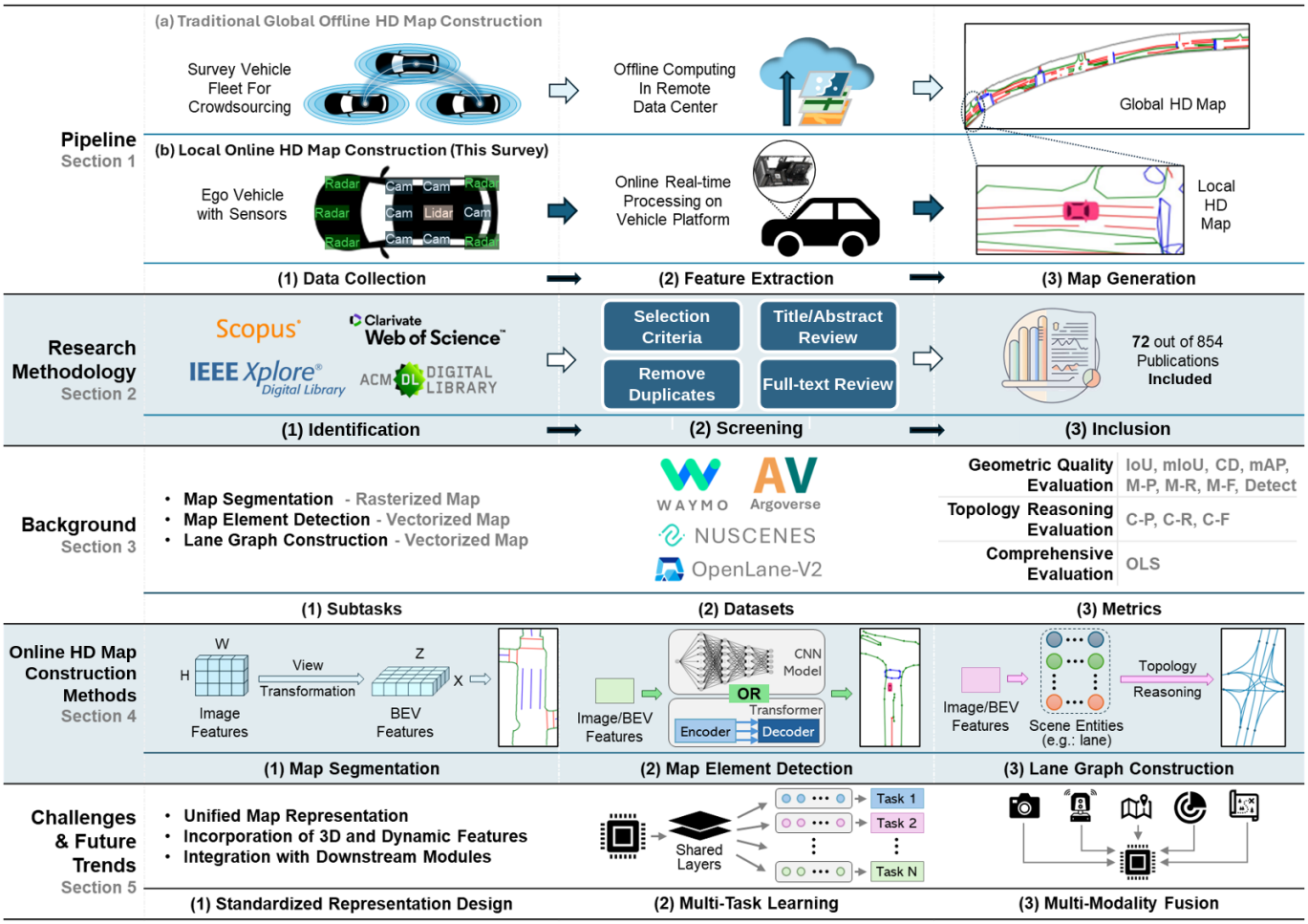


Figure 1. Structure of this survey.

2. Research Methodology

This section presents the research methodology for surveying online HD map construction, focusing on the search strategy, selection criteria, review process, and survey results.

2.1. Search Strategy

The authors conducted a literature search on online HD map construction across several scientific databases, including Scopus, IEEE Xplore, ACM Digital Library, and Web of Science. These databases cover fields such as computer science, electrical engineering, and intelligent transportation systems, ensuring a diverse range of academic sources. The search query used was: (((online OR construction OR reasoning) AND ((hd AND map) OR (lane AND graph) OR (lane AND topology))) AND ("autonomous vehicle" OR "autonomous vehicles" OR "autonomous driving")) OR ((bev OR "bird's eye view" OR "bird's-eye view" OR "bird's-eye-view") AND segmentation). Only papers published between January 2020 and January 2025 were included to ensure relevance and timeliness.

2.2. Selection Criteria

To ensure the relevance and quality of the papers in this survey, the authors applied inclusion and exclusion criteria to the articles obtained through the search strategy.

Inclusion criteria:

1. Focuses on online HD map construction;
2. Published between January 2020 and January 2025;
3. Published in peer-reviewed journals or conferences;

4. Written in English;
5. Full-text access available.

Exclusion criteria:

1. Research on offline HD map construction;
2. Duplicates;
3. Review, survey, data papers, book chapters, newsletters, or abstracts only;
4. Papers without experimental validation or contribution assessment;
5. Papers lacking comparison with existing literature.

2.3. Review Process

The authors used Rayyan [28], a web-based systematic review tool, to collaboratively screen literature from scientific databases. The process began with deduplication, arranging articles alphabetically, and reviewing title similarities to remove duplicates. The remaining articles were classified as 'INCLUDE', 'UNCERTAIN', or 'EXCLUDE' based on the title and abstract reviews, with reasons documented and disagreements resolved through discussion. Then a full-text review was conducted to refine the selection.

To expand the review, the authors used Google Scholar for citation tracing and reference tracking. Citation tracing uncovered studies missed in database searches, while reference tracking identified recent or unpublished works citing the selected studies.

The team independently analyzed and synthesized the selected literature, relying on their collective expertise without external input. The authors systematically reviewed the objectives, method, findings, and relevance of each study to the research focus. Regular discussions integrated diverse perspectives and insights from the literature, while mutual feedback and consensus building minimized biases in research findings.

2.4. Survey Results

Figure 2 presents the research methodology pipeline. The process began by retrieving relevant papers from multiple databases using search queries, yielding 854 records: 443 from Scopus, 116 from IEEE Xplore, 16 from ACM Digital Library, and 279 from Web of Science. The authors then performed a systematic screening to refine the selection. Specifically, 147 records were excluded because they did not meet the selection criteria, 242 were identified as duplicates, 247 were considered less relevant based on the review of the title and abstract, and 19 were excluded due to a lack of access to full text. After this screening, 199 records were subjected to a full-text review. Additionally, citation tracking and reference tracking were performed, resulting in the inclusion of 27 additional papers. In the end, 72 papers were finalized for inclusion in this survey study.

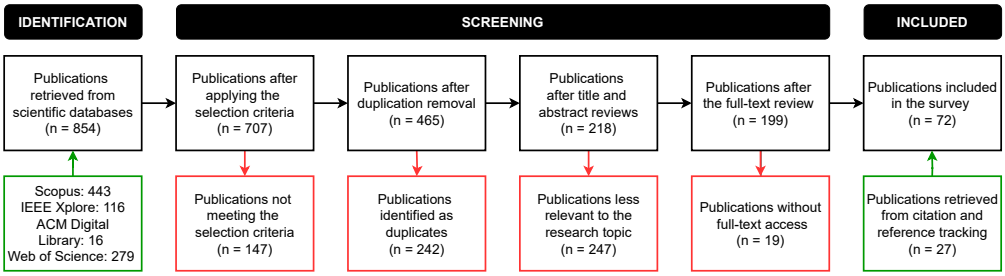


Figure 2. Pipeline of research methodology.

3. Background

This section provides background knowledge on online HD map construction. For a comprehensive overview, we cover task definitions, commonly used datasets, and prevalent evaluation metrics.

3.1. Task Definitions

We defined three sub-tasks in online HD map construction: **Map Segmentation** methods produce rasterized maps. In contrast, **Map Element Detection** and **Lane Graph Construction** methods generate vectorized maps.

Map Segmentation: This task leverages sensor observation data as input to dynamically create a rasterized map $\mathcal{M}_r \in \mathbb{R}^{H \times W}$, which depicts the surrounding road environment of the ego vehicle as a grid of map pixels. Each pixel $\mathcal{M}_r^{i,j} \in \mathcal{M}_r$ corresponds to a square area in the BEV and is assigned a map semantic category C that identifies the type of traffic control element present at that location. Here, H and W denote the height and width of the grid, respectively, while i and j represent spatial indices for each map pixel. This representation offers the ego vehicle the most fine-grained semantic and geometric information, allowing it to navigate and comply with traffic regulations.

Map Element Detection: This task leverages sensor observation data to dynamically create a vectorized map $\mathcal{M}_v = \{\mathcal{M}_v^i \mid i = 1, 2, ..., N_v\}$, detecting map elements within the ego vehicle's surrounding road environment. Each element is represented by an ordered point sequence S to capture its geometric attributes and assigned a map semantic category C. Here, N_v denotes the total number of vectorized map elements within $\mathcal{M}_v$. This representation provides the ego vehicle with instance-level semantic and geometric information for each map element, serving as an efficient sparse representation of the surrounding road environment.

Lane Graph Construction: This task leverages sensor observation data to dynamically create a vectorized map $M_v = \{V, E\}$, depicting the ego vehicle's surrounding road environment as a directed lane graph. This lane graph comprises a set of vertices $V = \{V_i \mid i = 1, 2, ..., N_v\}$ and directed edges $E\{V_i \mid i = 1, 2, ..., N_e\}$, with each edge connecting two vertices and indicating a direction from one vertex to another. Here, N_v and N_e denote the number of vertices and edges, respectively. The representation of the lane graph varies based on specific requirements, particularly regarding vertex and edge definitions. We will now introduce three common representations:

- *Representation 1*: Each vertex represents a point on the lane graph, defined by a vectorized sequence S_v that encodes its coordinates and attributes. Each edge is a directed line segment connecting two vertices and is characterized by an ordered point sequence S_p describing its geometric shape.
- *Representation 2*: Each vertex represents a lane, capturing its geometric shape with an ordered point sequence S. The edges are represented by an adjacency matrix I, where $I[V_i, V_j] = 1$ indicates that lane V_i connects to lane V_j, with the termination of lane V_j aligned to the beginning of lane V_i. Here, i and j are the indices of the lane vertices.
- *Representation 3*: Each vertex represents a lane or a traffic element, capturing its geometric shape with an ordered point sequence S. The edges are represented by two adjacency matrices: the first, I_{ll}, denotes the connectivity between lanes, where $I_{ll}[V_i, V_j] = 1$ indicates that lane V_i connects to lane V_j, with the termination of lane V_j aligned to the beginning of lane V_i. The second adjacency matrix, I_{lt}, describes the correspondence between lanes and traffic elements, where $I_{lt}[V_i, V_k] = 1$ signifies that lane V_i is related to traffic element V_k. Here, i and j are the indices of the lanes, and k is the index of the traffic elements.

3.2. Datasets

Table 1 shows common datasets for online HD map construction, among which nuScenes [29], Argoverse 2 [30], and Openlane-V2 [31] are the three most influential.

NuScenes Dataset. NuScenes [29] is a comprehensive multimodal autonomous driving dataset launched in 2019. It features 1000 driving scenes recorded in Singapore and Boston, each lasting 20 s. The dataset is divided into 700 scenes for training, 150 for validation, and 150 for testing. NuScenes is notable for being the first dataset to include a full suite of AV sensors: 6 cameras with a resolution of 1600 × 900, a 32-beam spinning LiDAR, and 5 radars with a detection range of up to 250 m, offering 360-degree environmental perception for AVs. In addition, NuScenes provides high-precision, human-annotated semantic maps in both rasterized and vectorized formats. The original rasterized maps cover only roads and sidewalks with a resolution of 10 px/m. In contrast, the vectorized maps offer more detailed information with 11 semantic classes, including road dividers, lane dividers, and pedestrian crossings. These maps are instrumental as strong priors for downstream tasks such as prediction and planning.

Table 1. Datasets for online HD map construction. Under **Region**, "AS" is Asia, and "NA" is North America. Under **Data**, "# Scenes" refers to the number of data segments, and "# Scans" represents the number of point clouds. Under **Map Annotation**, "# Layers" indicates the number of map semantic layers, and "3D" indicates whether the map is represented in 3D format. The symbol "–" denotes that the statistic is unavailable.

Dataset	Year	Region	Sensor Data			Map Annotation			
			# Scenes	# Images	# Scans	# Cities	# Layers	Resolution	3D
Waymo [32]	2019	NA	1150	12 M	230 k	6	7	–	✗
NuScenes [29]	2019	NA/AS	1000	1.4 M	390 k	2	11	10 px/m	✗
Argoverse 1 [33]	2019	NA	113	490 k	22 k	2	3	–	✓
Argoverse 2 [30]	2021	NA	1000	2.7 M	150 k	6	4	100 px/m	✓
OpenLane-V2 [31]	2023	NA/AS	2000	466 K	–	8	3	–	✓

Argoverse 2 Dataset. Argoverse 2 [30] is a comprehensive dataset for autonomous driving perception and prediction research, released in 2023 as an upgrade to the Argoverse [33] dataset. It features 1000 diverse driving scenarios from six U.S. cities, each consisting of a 15 s multimodal data sequence, divided into 700 scenarios for training, 150 for validation, and 150 for testing. The dataset has an advanced sensor suite that provides a full panoramic field of view, including seven cameras with 2048 × 1550 pixel resolution, two roof-mounted 32-beam spinning LiDARs, and two stereo cameras of the same resolution. It also provides detailed 3D local HD maps in vectorized and rasterized formats. The vectorized maps include lane-level details such as lane boundaries, markings, crosswalks, and drivable areas, while the rasterized maps offer dense ground surface height data. This 3D map representation enhances lane geometry details, aiding AVs in better perceiving surrounding static traffic infrastructure.

OpenLane-V2 Dataset. OpenLane-V2 [31] is the first dataset designed to advance HD map construction through topology reasoning of traffic scene structures, launched in 2023. Building upon the existing nuScenes and Argoverse 2 datasets, it includes 2000 diverse annotated road scenes divided into two subsets of 1000 scenes each, with 700 scenes for training, 150 for validation, and 150 for testing. The dataset offers detailed annotations featuring vectorized HD maps and traffic elements. HD maps provide comprehensive information on lane segments, including centerlines, boundaries, marking types, and lane connectivity. Traffic elements such as traffic lights, road markings, and road signs are annotated within 2D bounding boxes on front-view images, and their relationships to lanes are represented as adjacency matrices for each frame. These detailed annotations and their relationships improve AVs' comprehension of complex road environments, supporting more accurate navigation and decision-making.

3.3. Evaluation Metrics

Evaluating online HD map construction requires a comprehensive assessment framework due to its complex nature. Here, we group existing metrics into three categories: (1) Geometric Quality Evaluation measures the precision of spatial and geometric attributes; (2) Topology Reasoning Evaluation examines the connectivity and correspondence between map elements; and (3) Comprehensive Evaluation provides an overall performance assessment by integrating multiple metrics.

3.3.1. Geometric Quality Evaluation

Intersection over Union (IoU). The most common metric to evaluate pixel-level localization performance in detection and segmentation tasks is the Intersection over Union (IoU) [23,34]. It assesses the similarity between the predicted representation (D_P) and the ground-truth geometry (D_G) by calculating the ratio of the intersection of the two areas ($D_P \cap D_G$) to their combined areas ($D_P \cup D_G$). Accordingly, IoU is mathematically defined as:

$$\text{IoU}(D_P, D_G) = \frac{D_P \cap D_G}{D_P \cup D_G},\ D_P, D_G \in \mathbb{R}^{H \times W \times D}, \tag{1}$$

in which H and W refer to the height and width of the constructed map, respectively, and D corresponds to the category amount involved in the map. The value of IoU ranges from 0 to 1, where higher values signify better alignment.

Mean Intersection over Union (mIoU). Various improvements have been proposed to enhance the clarity of IoU. For instance, by taking the different semantic category of each detected or segmented map element into account, $mIoU$ is computed by averaging the IoU values across all classes. The corresponding mathematical definition is:

$$mIoU = \frac{1}{D} \sum_{d=1}^{D} \text{IoU}_d, \tag{2}$$

where D is the number of classes considered in the map.

Chamfer Distance (CD). CD is a Lagrangian metric that quantifies the spatial similarity between two vector geometric shapes, the predicted curve C_P and the ground-truth one C_G. It computes the average minimum squared distances between the constituent points on C_P and C_G, in both forward and backward directions [35]. Specifically, CD_P refers to the directional calculation from prediction to ground-truth, while CD_G denotes the directional calculation from ground-truth to prediction [35]. Hence, CD is mathematically defined as:

$$CD_P(C_P, C_G) = \frac{1}{P} \sum_{p \in P} \min_{g \in G} p - g_2, \tag{3}$$

$$CD_G(C_G, C_P) = \frac{1}{G} \sum_{g \in G} \min_{p \in P} g - p_2, \tag{4}$$

$$CD = CD_P(C_P, C_G) + CD_G(C_G, C_P), \tag{5}$$

in which P and G represent the sets of points sampled on C_P and the C_G, respectively.

Mean Average Precision (mAP). mAP serves as a classical metric to evaluate the accuracy of HD map construction [23]. It quantifies the true positives in predictions compared to the ground truths. Specifically, while conceptually similar to AP in 2D object detection, AP in HD map construction adopts CD or Frechet distance rather than IoU as its matching criterion. Only when the corresponding distance is smaller than the threshold t will the prediction be conceived as a true positive [23]. This adaptation accounts for the vector-based, geometric nature of HD map elements compared to pixel-based 2D images.

Further, the threshold usually takes a value from the set $T = \{0.5, 1.0, 1.5\}$ [23]. Then, mAP is computed by averaging the AP computed across all adopted thresholds. Hence, the corresponding mathematic definition is written as:

$$mAP = \frac{1}{T} \sum_{t \in T} AP_t. \tag{6}$$

Centerline Identification Metrics (M-P, M-R, M-F, Detect). Similar to 2D classification tasks, fundamental evaluation metrics, precision, recall, and F1-score, have also been adapted to assess the accuracy of centerline identification in lane graph extraction. Specifically, these adapted metrics, denoted as *M-P*, *M-R*, and *M-F*, evaluate how closely predicted centerlines align with the ground truths within a predefined distance threshold [19,36]. Meanwhile, since these metrics do not penalize cases where a ground truth centerline lacks any matching prediction, the Detection Ratio (*Detect*) is proposed. It measures the fraction of ground truth centerlines that have at least one matching estimated centerline [19,36]. Hence, a low *Detect* score combined with high scores on the other three metrics indicates a significant number of false negatives despite a satisfying performance in identifying true positives.

3.3.2. Topology Reasoning Evaluation

Connectivity Metrics (C-P, C-R, C-F). Given the crucial role of relation in graph interpretation, connectivity metrics are proposed to assess the accuracy of edge construction in the lane graph. These metrics, *C-P*, *C-R*, and *C-F*, are also adapted from standard precision and recall, while focusing on how well the connectivity pattern of the predicted graph complies with that of the ground-truth graph [19,36].

3.3.3. Comprehensive Evaluation

OpenLane-V2 Score (*OLS*). OLS incorporates four metrics to comprehensively evaluate the performance across three main subtasks supported by the OpenLane-V2 dataset [31]. Specifically, DET_l, as a modified mAP metric, assesses 3D lane detection performance, where Frechet distance serves as the matching criterion with the threshold set $\mathbb{T} = \{1.0, 2.0, 3.0\}$ meters. Similarly, DET_t appraises traffic element detection performance while using $(1 - IoU)$ with a threshold of 0.75 as the affinity measure to better align with the small-scaled nature of traffic elements. Further, TOP_{ll} and TOP_{lt} evaluate topology reasoning performance among centerlines or between lane centerlines and traffic elements, respectively. Hence, OLS is mathematically defined as:

$$\text{OLS} = \frac{1}{4}\left[\text{DET}_l + \text{DET}_t + f(\text{TOP}_{ll}) + f(\text{TOP}_{lt})\right] \tag{7}$$

where f is a scale function that weights the topology reasoning task.

4. Online HD Map Construction Methods

This section explores various perspectives on online HD map construction. We categorize the methods into three sub-tasks based on the output map format: Map Segmentation in Section 4.1, Map Element Detection in Section 4.2, and Lane Graph Construction in Section 4.3.

Table 2 offers an overview of existing research on online HD map construction, aiming to help readers quickly understand the evolution and innovations of different methods. It organizes the studies into three sections based on the specific sub-tasks and provides key details, including venue, sensor modality, task, dataset, and contributions. Each entry highlights major contributions, such as new network architectures, view transformation

techniques, or sensor fusion strategies. The table shows a growing trend in publications at top conferences and journals, indicating the research field's rapid development and increasing importance.

Table 2. Literature on online HD map construction. Under **Modality**, "SC" is single-camera, "MC" is multi-camera, and "L" is LiDAR. Under **Task**, "MapSeg" means map segmentation, "MapEle" means map element detection, and "LaneGra" is for lane graph construction. Under **Dataset**, "nuS" is for nuScenes [29], "AV" is for Argoverse [33], "K360" is for Kitti-360 [37], "AV" is for Argoverse 2 [30], and "OL2" is for OpenLane-V2 [31].

Method	Venue	Modality	Task	Dataset	Contribution
PON [38]	*CVPR 2020*	SC	MapSeg	nuS/AV	MLP for Depth Axis Expansion
LSS [39]	*ECCV 2020*	MC	MapSeg	nuS	CNN for Pixel-Wise Depth Prediction
Cam2BEV [40]	*ITSC 2020*	MC	MapSeg	Synthetic	IPM for VT on 2D Feature Maps
VPN [41]	*RA-L 2020*	MC	MapSeg	Synthetic	MLP to Learn Projection for VT
PYVA [42]	*CVPR 2021*	SC	MapSeg	AV	Cycled View Projection for VT
STA-ST [43]	*ICRA 2021*	SC	MapSeg	nuS	Temporal Fusion After VT
CVT [44]	*CVPR 2022*	MC	MapSeg	nuS	Camera-Aware Embedding to Enhance VT
BEVFormer [45]	*ECCV 2022*	MC	MapSeg	nuS	Transformer and Projection for VT
Ego3RT [46]	*ECCV 2022*	MC	MapSeg	nuS	Ego 3D Representation to Enhance VT
LaRa [47]	*CoRL 2022*	MC	MapSeg	nuS	Ray Embedding to Enhance VT
PanopticSeg [48]	*RA-L 2022*	SC	MapSeg	nuS/K360	Hybrid VT with IPM and Depth Expansion
M^2BEV [49]	*arXiv 2022*	MC	MapSeg	nuS	3D Voxel Grid Projected onto 2D Features
BEVerse [50]	*arXiv 2022*	MC	MapSeg	nuS	Spatio-Temporal Fusion after VT
BEVSegFormer [51]	*WACV 2023*	MC	MapSeg	nuS	Transformer to Learn Projection for VT
BEVFusion [52]	*ICRA 2023*	MC/L	MapSeg	nuS	Camera–Lidar Fusion on BEV after VT
Simple-BEV [53]	*ICRA 2023*	MC/L/R	MapSeg	nuS	Bilinear Sampling for Voxel Quality
HFT [54]	*ICRA 2023*	SC	MapSeg	nuS/AV	Mutual Learning to Enhance Hybrid VT
PETRv2 [55]	*ICCV 2023*	MC	MapSeg	nuS	3D Position Embedding to Enhance VT
HDMapNet [21]	*ICRA 2022*	MC/L	MapEle	nuS	FCN with Post-Processing for MD
VectorMapNet [22]	*ICML 2023*	MC/L	MapEle	nuS/AV2	Transformer for Vectorized MD
MapTR [56]	*ICLR 2023*	MC	MapEle	nuS	Single-Stage Transformer for Parallel MD
BeMapNet [57]	*CVPR 2023*	MC	MapEle	nuS	Piecewise Bezier Curve for MD
NMP [58]	*CVPR 2023*	MC	MapEle	nuS	Global Neural Map Prior for MD
InstaGraM [59]	*CVPRW 2023*	MC	MapEle	nuS	CNNs and GNN for Graph-Based MD
PivotNet [60]	*ICCV 2023*	MC	MapEle	nuS/AV2	Pivot-Based Representation for MD
ScalableMap [61]	*CoRL 2023*	MC	MapEle	nuS	Hierarchical Sparse Map Representation
MapVR [62]	*NeurIPS 2023*	MC/L	MapEle	nuS/AV2	Rasterization for Geometric Supervision
StreamMapNet [63]	*WACV 2024*	MC	MapEle	nuS/AV2	Streaming Temporal Fusion for MD
DPFormer [64]	*AAAI 2024*	MC	MapEle	nuS	Douglas–Peucker Point Representation
SuperFusion [65]	*ICRA 2024*	MC/L	MapEle	nuS	Multi-Level LiDAR–Camera Fusion for MD
HIMap [66]	*CVPR 2024*	MC	MapEle	nuS/AV2	Integrated HIQuery for MD
MGMap [67]	*CVPR 2024*	MC/L	MapEle	nuS/AV2	Mask-Guided Learning for MD
InsMapper [68]	*ECCV 2024*	MC/L	MapEle	nuS/AV2	Inner-Instance Information for MD
GeMap [69]	*ECCV 2024*	MC	MapEle	nuS/AV2	Geometric Invariant Representation
HRMapNet [70]	*ECCV 2024*	MC	MapEle	nuS/AV2	Global Historical Rasterized Map for MD
MapDistill [71]	*ECCV 2024*	MC	MapEle	nuS	Cross-Modal Knowledge Distillation
SQD-MapNet [72]	*ECCV 2024*	MC	MapEle	nuS/AV2	Stream Query Denoising for Consistency
MapTracker [73]	*ECCV 2024*	MC	MapEle	nuS/AV2	Strided Memory Fusion for MD
ADMap [74]	*ECCV 2024*	MC/L	MapEle	nuS/AV2	Anti-Disturbance MD Framework
MapQR [75]	*ECCV 2024*	MC	MapEle	nuS/AV2	Scatter-and-Gather Query for MD
MapTRv2 [76]	*IJCV 2024*	MC/L	MapEle	nuS/AV2	Advanced Baseline Method for MD
DTCLMapper [77]	*T-ITS 2024*	MC	MapEle	nuS/AV2	Dual-Stream Temporal Consistency Learning
P-MapNet [78]	*RA-L 2024*	MC/L	MapEle	nuS/AV2	SD and HD Map Priors for MD
PrevPredMap [79]	*WACV 2025*	MC	MapEle	nuS/AV2	Temporal Fusion with Previous Predictions

Table 2. *Cont.*

Method	Venue	Modality	Task	Dataset	Contribution
STSU [19]	*ICCV 2021*	SC	LaneGra	nuS	MLP to Infer Lane Connectivity
TPLR [20]	*CVPR 2022*	SC	LaneGra	nuS/AV	Minimal Cycles for Lane Graph Topology
CenterLineDet [80]	*ICRA 2023*	MC	LaneGra	nuS	Transformer for Iterative TR
VideoLane [81]	*ITSC 2023*	SC	LaneGra	nuS/AV	Temporal Aggregation to Enhance TR
LaneWAE [82]	*ITSC 2023*	SC	LaneGra	nuS/AV	Dataset Prior Distribution to Enhance TR
ObjectLane [83]	*ICCV 2023*	SC	LaneGra	nuS/AV	Object–Lane Clustering to Enhance TR
RoadNetTransformer [84]	*ICCV 2023*	MC	LaneGra	nuS	RoadNet Sequence Representation for TR
TopoNet [85]	*arXiv 2023*	MC	LaneGra	OL2	GNN to Refine Lane Graph Topology
LaneGraph2Seq [36]	*AAAI 2024*	MC	LaneGra	nuS/AV2	Graph Sequence Representation for TR
LaneSegNet [86]	*ICLR 2024*	MC	LaneGra	OL2	Lane Segment Representation for TR
TopoMLP [87]	*ICLR 2024*	MC	LaneGra	OL2	Robust Detectors to Enhance TR
SMERF [88]	*ICRA 2024*	MC	LaneGra	OL2	SD Maps to Enhance TR
LaneMapNet [89]	*IV 2024*	MC	LaneGra	nuS	Curve Region-Aware Attention
CGNet [90]	*ECCV 2024*	MC	LaneGra	nuS/AV2	GNN and GRU to Optimize Lane Graph
RoadPainter [91]	*ECCV 2024*	MC	LaneGra	OL2	Points-Mask Optimization
LaneGAP [92]	*ECCV 2024*	MC	LaneGra	nuS/AV2/OL2	Post-Processing to Restore Lane Topology
TopoMaskV2 [93]	*ECCVW 2024*	MC	LaneGra	OL2	Mask-Based Formulation to Enhance TR
TopoLogic [94]	*NeurIPS 2024*	MC	LaneGra	OL2	Lane Geometry and Query Similarity for TR

4.1. Map Segmentation for Rasterized Maps

Map segmentation algorithms dynamically generate rasterized maps representing the road environment around the ego vehicle as a pixel grid with semantic information. In this semantic understanding process, the camera captures visual details such as color, texture, and shape [95,96]. A key challenge in converting images to rasterized maps is that the input and output exist in different spaces: the former in the image plane and the latter in the BEV plane. To address this, mainstream vision-based map segmentation algorithms propose various view transformation (VT) methods to convert images or features from the 2D image plane to the 3D world space, decoding rasterized maps in the BEV.

Recent research [38,39,44,45,48] has focused on enhancing this VT module, leading us to categorize map segmentation methods into three main types based on VT techniques. The first category, "**Projection-based Methods**", employs the projection model defined by the camera's intrinsic and extrinsic parameters to implement VT. The second category, "**Lift-based Methods**", involves a VT module that elevates images or features to 3D space by recovering depth information. The third category, "**Network-based Methods**", achieves VT implicitly through neural networks.

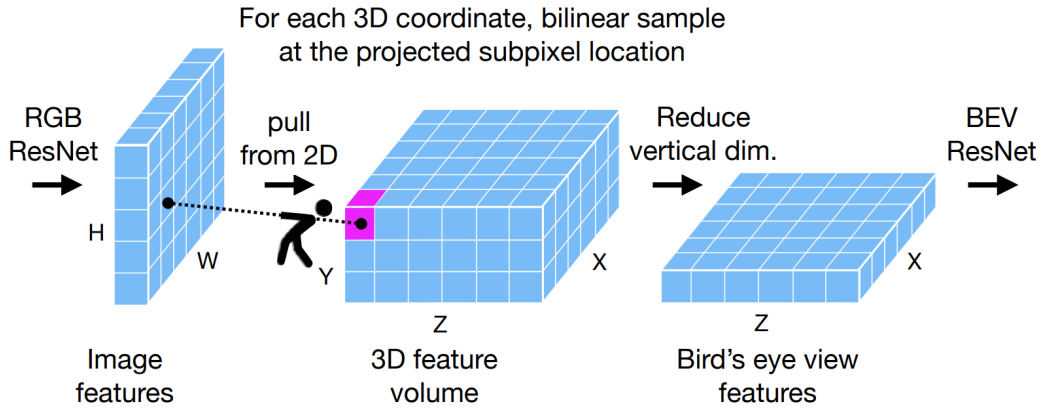
4.1.1. Projection-Based VT for Map Segmentation

Projection-based methods utilize the camera projection model for VT to generate rasterized maps. These methods trace back to inverse perspective mapping (IPM) [97], which converts perspective images into BEV images to eliminate distortion. It assumes that inverse-mapped points lie on a horizontal reference plane and employs the camera's intrinsic and extrinsic parameters for converting pixels to world coordinates. While effective for preprocessing image-like data [98–101], IPM can introduce distortions when handling 3D objects. To address this, Cam2BEV [40] employs deep learning to correct these distortions. The method applies IPM for VT on feature maps from various cameras, followed by deep learning refinement to enhance the accuracy of the rasterized maps.

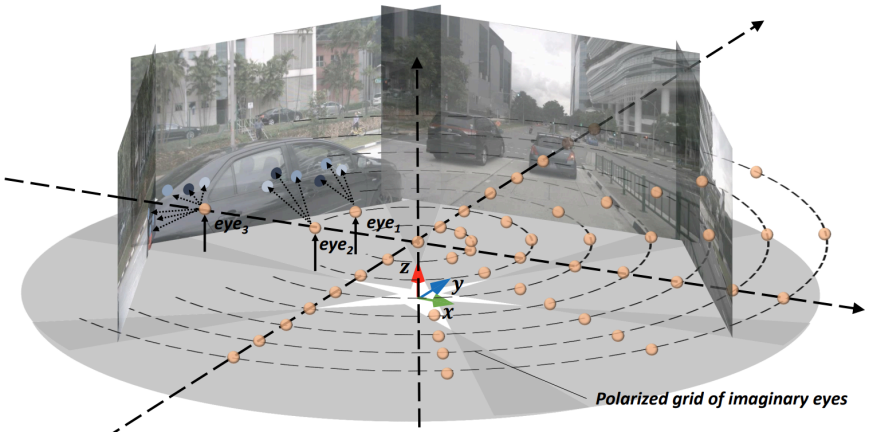
Other methods enhance the quality of rasterized maps by constructing 3D voxel features that retain high-dimensional information during VT. They initialize a voxel grid in world space and populate it with 2D feature maps, guided by the projection relationships defined by the camera's intrinsic and extrinsic parameters. For example, M^2BEV [49] assumes a uniform depth distribution, filling the depth-direction voxels along the camera

rays with corresponding 2D features, which are then height-compressed and decoded into a rasterized map. Simple-BEV [53], as depicted in Figure 3a, improves voxel quality by using bilinear sampling on the feature map for more effective grid filling.

Inspired by Transformer architecture, methods combining camera projection relationships and cross-attention for constructing 3D voxel features have recently gained traction. BEVFormer [45] introduces a spatial cross-attention mechanism to generate BEV features adaptively. It first projects predefined grid-like BEV queries onto the 2D camera view, then uses a deformable attention mechanism [102] for interaction with sampled features in the regions of interest, and finally aggregates multi-view features to decode them into a rasterized map. Similarly, Ego3RT [46] (Figure 3b) introduces a multi-view adaptive attention mechanism that dynamically extracts key features from multi-view feature maps to generate 3D voxel features. By constructing BEV queries in a polarized coordinate system, this approach better aligns with the geometric distribution of the ego vehicle's surroundings. F2BEV [103] introduces a distortion-aware spatial cross-attention mechanism to generate BEV features from fisheye images. It employs a unified projection model derived from fisheye camera parameters to correct radial and tangential distortions, accurately projecting 3D reference points from the BEV plane to the fisheye camera's 2D views.



(**a**)



(**b**)

Figure 3. Comparison of the VT module in two projection-based map segmentation methods. (**a**) Simple-BEV [53] projects voxel grid points onto feature maps and uses bilinear sampling to extract features for constructing 3D voxel features. (**b**) Ego3RT [46] projects polarized grid queries onto feature maps and uses attention to extract features for constructing 3D voxel features.

4.1.2. Lift-Based VT for Map Segmentation

Lift-based methods generate rasterized maps by recovering depth information from images for VT. Initially proposed by PON [38] (Figure 4a), this approach employs dense transformation layers made of multi-layer perceptrons (MLP). It begins with compressing feature maps along the height axis, then unfolds bottleneck features along the depth axis to create BEV features in polar coordinates. These features are resampled into an orthogonal coordinate system using camera parameters and then decoded into rasterized maps. ViT-BEVSeg [104] adopts the hierarchical vision Transformer blocks and introduces a vision Transformer as the backbone. Similarly, STA-ST [43] adopts this VT methodology and enhances it with temporal information and multi-scale supervision to improve map quality. BEVStitch [105] also uses temporal information by mapping features from different time frames to a common BEV coordinate system and aggregating them into a unified feature map.

Other methods utilize convolutional neural networks (CNNs) to recover depth information from images at a granular level, enhancing the quality of rasterized maps. LSS [39], as shown in Figure 4b, was the first to use CNNs to predict depth probability distributions and contextual features for each pixel in the feature map. These are combined using an outer product to create frustum features with depth information. Finally, these frustum features are projected onto a unified BEV grid for fusion and decoded into a rasterized map. Subsequent methods adopted a similar VT approach, enhanced through deep supervision, spatiotemporal fusion, sensor fusion, and multitask learning. BEVerse [50] aligns past BEV features with current ones, feeding them into decoders for three distinct tasks to improve joint learning and optimization efficiency. In contrast, BEVFusion [52] integrates data from multi-view cameras and LiDAR, enabling the fused BEV features to support both 3D object detection and BEV map segmentation tasks effectively.

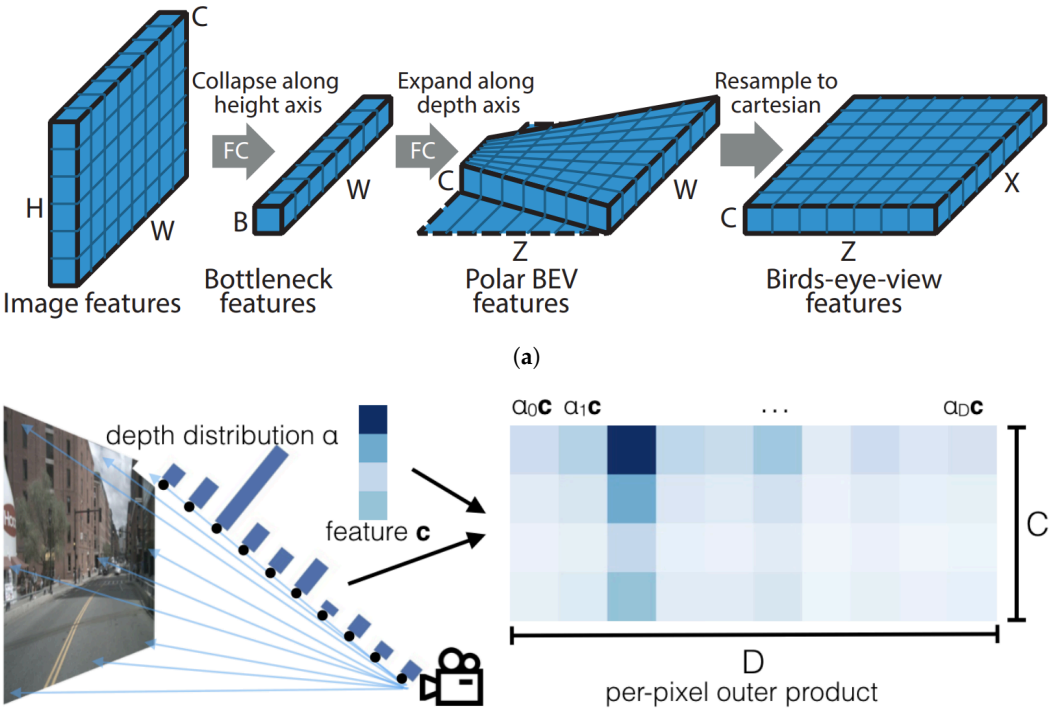
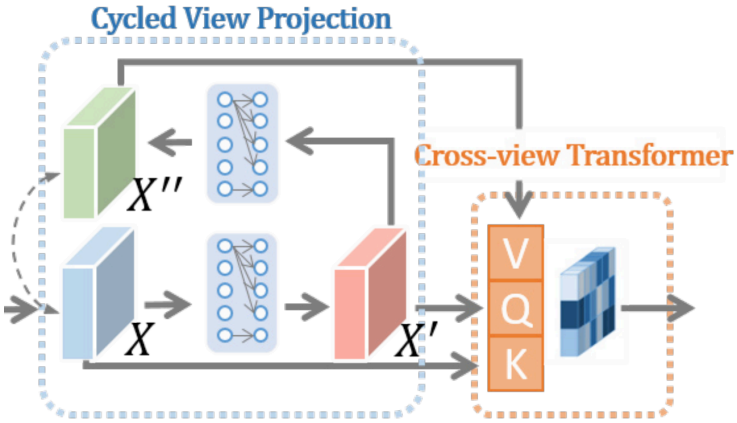


Figure 4. Comparison of the VT module in two lift-based map segmentation methods. (**a**) PON [38] uses MLP to expand bottleneck features along the depth axis. (**b**) LSS [39] uses CNN to predict pixel-wise depth probability distributions.

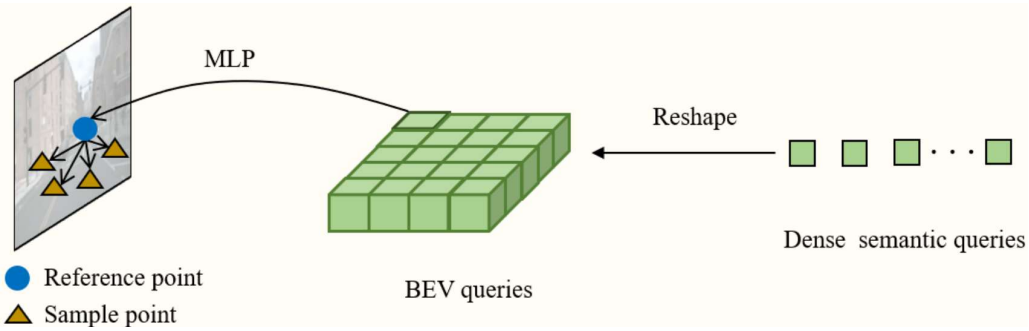
4.1.3. Network-Based VT for Map Segmentation

Network-based methods achieve VT by implicitly encoding camera projection relationships with neural networks. They originate from VPN [41], which uses an MLP to learn spatial dependencies between pixel and BEV coordinate systems. The process involves mapping feature maps to BEV space with a two-layer MLP and fusing multi-view feature maps to decode a rasterized map. However, the significant perspective differences between the perspective view and BEV can lead to information loss during direct feature mapping. To align features before and after mapping, PYVA [42] (Figure 5a) introduced cycled view projection. This method uses two MLPs for bidirectional transformations between the two spaces and employs cyclic consistency loss to ensure feature coherence during the conversion process.

Other methods utilize the Transformer architecture to capture the camera projection relationship, significantly improving the quality of rasterized maps. Tesla [106] was the first to implement the cross-attention mechanism from this architecture to model the projection relationship. This approach constructs dense BEV queries and uses cross-attention to refine these queries through interaction with feature maps, ultimately decoding outputs for various perception tasks. However, the computational cost of traditional cross-attention increases quadratically with input size, leading to significant overhead when processing high-dimensional data. To address this, BEVSegFormer [51] (Figure 5b) utilizes deformable cross-attention [102], a sparse variant that generates 2D reference points for each BEV query using an MLP. It then dynamically samples the nearby regions of feature maps to refine the BEV queries for decoding the rasterized map.



(**a**)



(**b**)

Figure 5. Comparison of the VT module in two network-based map segmentation methods. (**a**) PYVA [42] uses two MLPs to enable bidirectional projection of feature maps between pixel space and BEV space. (**b**) BEVSegFormer [51] uses deformable cross-attention [102] to predict 2D reference points for sampling feature maps to refine BEV queries.

Various methods integrate camera geometric information into image features during VT to enhance the model's ability to capture geometric correspondences and improve cross-view fusion. CVT [44] pioneers camera-aware positional embeddings, converting pixel coordinates into 3D direction vectors that link geometric information across camera views. LaRa [47] follows a similar approach with ray embeddings. In contrast, PETRv2 [55] employs 3D position embeddings to enhance image features. The method maps points from the camera's frustum space to the 3D space. The 3D coordinates of these points are utilized to generate positional embeddings that integrate with image features, enhancing the model's perception of three-dimensional object information.

4.1.4. Discussion on Map Segmentation Methods

Map segmentation methods can be classified into three types based on VT techniques, each with unique advantages and limitations. First, projection-based methods explicitly establish the coordinate transformation between pixels and BEV space using a camera projection model. While they offer strong interpretability through clear geometric frameworks, their performance heavily relies on the accuracy of camera parameters, which can be affected by factors like vibration and temperature changes during dynamic driving. Second, lift-based methods recover lost depth information from the camera to elevate images into 3D space. Although they provide an intuitive VT solution, they depend on the accuracy of depth estimation and often require additional sensors, such as stereo cameras or LiDAR, for improved depth precision. Lastly, network-based methods employ neural networks to learn camera projection relationships for VT implicitly. This straightforward approach reduces reliance on precise camera parameters but tends to converge more slowly and requires substantial data to achieve optimal performance. Therefore, an important research direction is effectively leveraging the complementary advantages of different VT techniques to enhance map segmentation performance.

Some methods combine various VT techniques to enhance the quality of rasterized maps. PanopticSeg [48] introduces an architecture with two VT modules for handling vertical and horizontal regions. The horizontal Transformer uses IPM and an error correction module, while the vertical Transformer lifts feature maps into a 3D voxel grid and resamples them using camera intrinsic parameters. This approach fuses the BEV features from both modules to produce the rasterized map. HFT [54] later adopts a similar dual-stream architecture, incorporating a mutual learning mechanism that enables the Transformers to share information through feature imitation, improving overall accuracy.

Table 3 shows the results of map segmentation methods on the nuScenes [29] validation set, revealing two key observations. First, network-based methods significantly underperform other methods. For instance, in the single-camera setting, the mIoU of VED [107] and VPN [41] is only 25.2% and 31.8%, respectively. In the multi-camera setting, CVT [44] also performs poorly with an mIoU of just 40.2%. This may result from network-based methods not explicitly utilizing camera geometric information during the VT process. Second, incorporating LiDAR point clouds significantly enhances vision-based methods. For example, BEVFusion [52] saw a 6.1% increase in mIoU in a multi-camera setup after integrating LiDAR data. This improvement is likely due to the complementary information from cameras and LiDAR: cameras provide rich color and texture, while LiDAR offers precise distance measurements.

Table 3. Performance comparison of map segmentation methods on the nuScenes [29] validation set. Under **Modality**, "SC" is for single-camera, "MC" is multi-camera, and "L" is for LiDAR. Under **VT Type**, "Projection", "Lift", and "Network" denote three types of map segmentation methods based on view transformation techniques. The symbol * indicates results from [43], † denotes results from [52], and "–" denotes that the statistic is unavailable.

Method	Modality	VT Type	Drivable	Ped. Cross.	Walkway	Stop Line	Carpark	Divider	mIoU
VED * [107]	SC	Network	54.7	12.0	20.7	–	13.5	–	25.2
VPN * [41]	SC	Network	58.0	27.3	29.4	–	12.3	–	31.8
PON * [38]	SC	Lift	60.4	28.0	31.0	–	18.4	–	34.5
STA-ST * [43]	SC	Lift	71.1	31.5	32.0	–	28.0	–	40.7
CVT † [44]	MC	Network	74.3	36.8	39.9	25.8	35.0	29.4	40.2
OFT † [108]	MC	Projection	74.0	35.3	45.9	27.5	35.9	33.9	42.1
LSS † [39]	MC	Lift	75.4	38.8	46.3	30.3	39.1	36.5	44.4
M^2BEV † [49]	MC	Projection	77.2	–	–	–	–	40.5	–
BEVFusion† [52]	MC	Lift	81.7	54.8	58.4	47.4	50.7	46.4	56.6
BEVFusion † [52]	MC & L	Lift	85.5	60.5	67.6	52.0	57.0	53.7	62.7

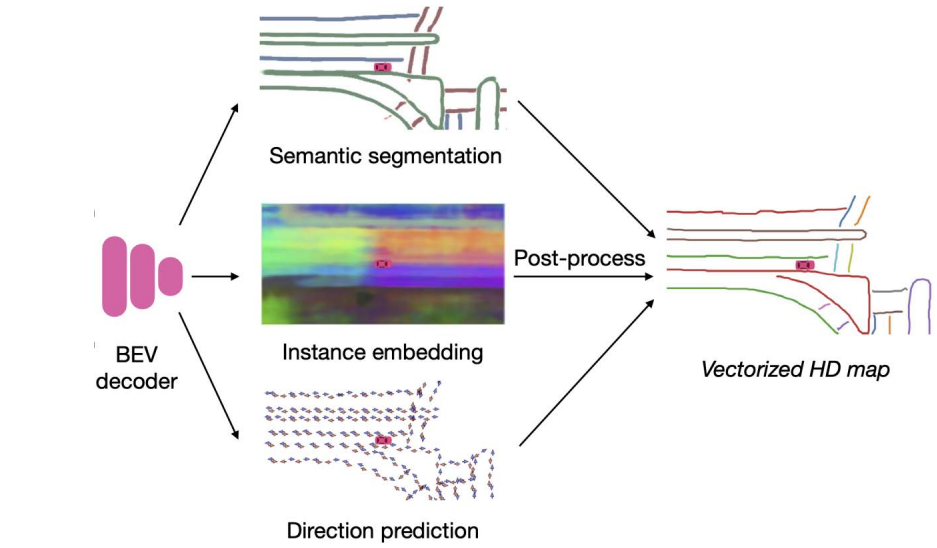
4.2. Map Element Detection for Vectorized Maps

Map element detection algorithms dynamically generate a vectorized map that includes traffic-related elements in the surrounding road environment, providing detailed geometric and semantic information. A key challenge in this process is detecting and classifying diverse map elements as ordered point sequences with semantic categories. This requires the algorithm to extract geometric shapes and classifications while managing occlusions from vehicles and pedestrians under varying lighting and weather conditions. To tackle this challenge, leading map element detection algorithms propose various map decoding (MD) techniques to transform image or BEV features into precise vectorized representations, reconstructing the surrounding environment's structure and semantics.

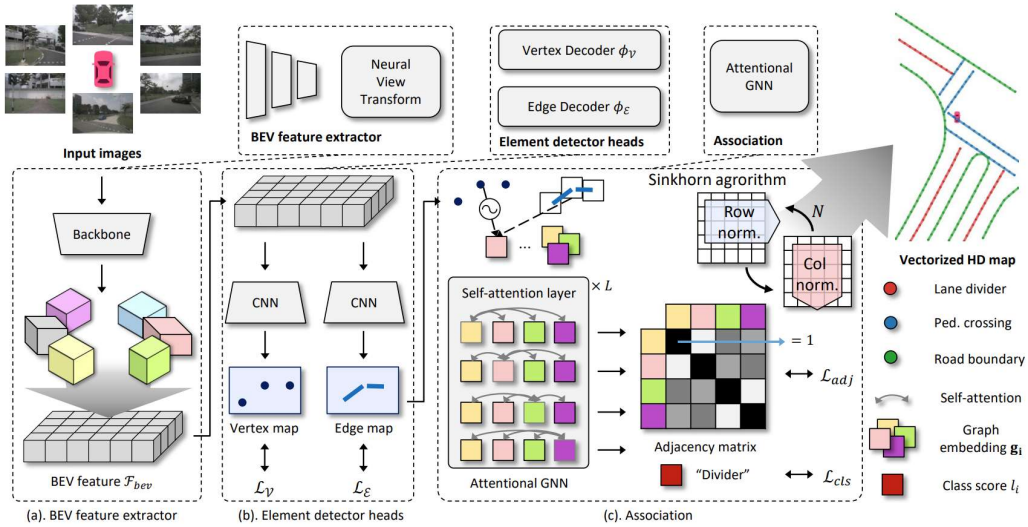
Recent research [21,22,56,63] has focused on enhancing this MD module, enabling the classification of map element detection methods into two main categories based on the decoder architecture. The first category, "**CNN-based methods**", employs CNN to process image or BEV features hierarchically, capturing local details and spatial structures for accurate map element detection. The second category, "**Transformer-based methods**", leverages the Transformer's self-attention mechanisms to capture long-range dependencies and global context in image or BEV features, enabling precise map element detection in complex road environments.

4.2.1. CNN-Based MD for Map Element Detection

CNN-based methods utilize CNN to process image or BEV features for detecting map elements. These methods were first introduced by HDMapNet [21] (Figure 6a), which uses a fully convolutional network [109] to detect map elements. Its decoder features three branches for semantic segmentation, instance embedding, and direction prediction. The outputs from the decoder are then post-processed through clustering and non-maximum suppression (NMS), resulting in accurate and well-structured map elements. Building on this, SuperFusion [65] adopts a similar decoding approach and introduces a multi-level LiDAR–camera fusion mechanism to enhance long-range detection. The method includes data-level fusion using LiDAR depth to enhance images, feature-level fusion where image features guide LiDAR's BEV features, and BEV-level fusion that aligns and merges BEV features from both modalities.



(**a**)



(**b**)

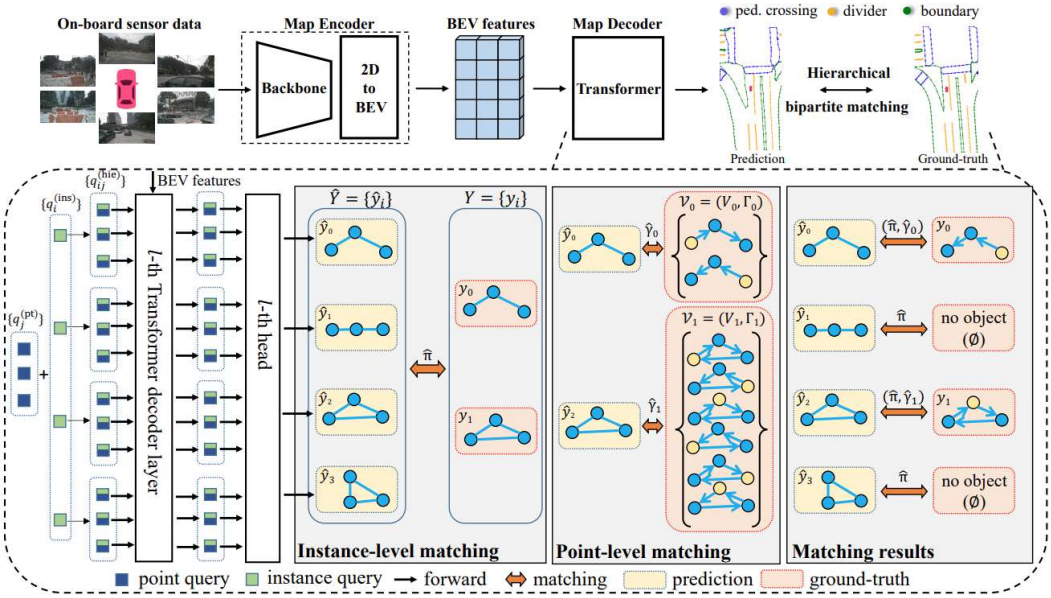
Figure 6. Comparison of the MD module in two CNN-based map element detection methods. (**a**) HDMapNet [21] uses an FCN [109] to decode semantic, instance, and direction masks, which are then post-processed into vectorized representations. (**b**) InstaGraM [59] uses two CNNs to detect vertices and edges, then employs an attentional GNN to associate the vertices, generating vectorized representations in an end-to-end manner.

Some methods have introduced end-to-end CNN decoding strategies that streamline workflows and improve map element detection accuracy by reducing post-processing steps. InstaGraM [59], as depicted in Figure 6b, combines CNNs and a graph neural network (GNN) to extract and relate map elements. The method uses two CNNs to detect vertices and edges, then employs an attentional GNN to associate the vertices, ultimately generating vectorized representations that conform to road topology.

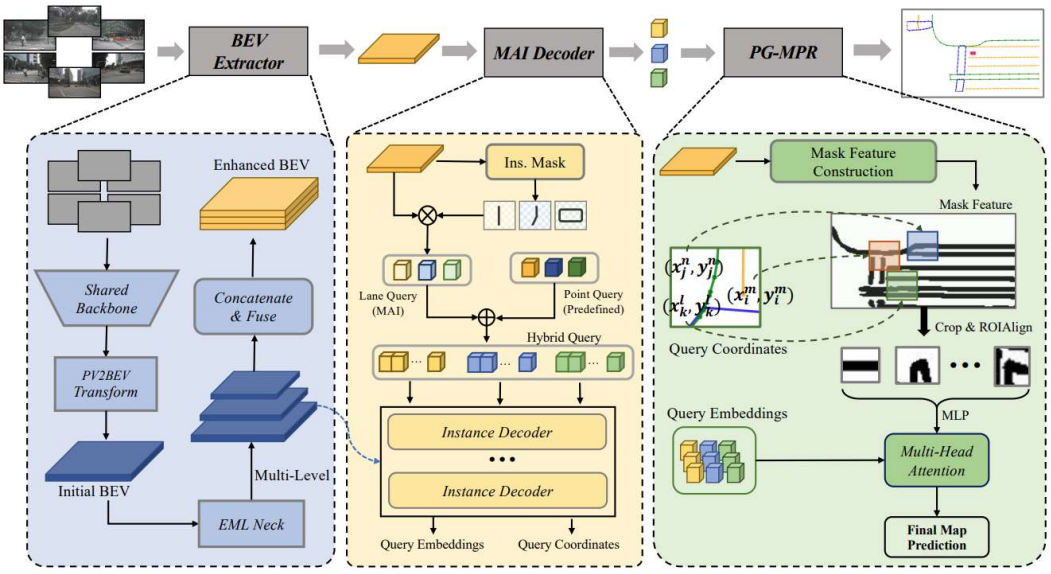
4.2.2. Transformer-Based MD for Map Element Detection

Transformer-based methods leverage Transformer's self-attention mechanisms to capture long-range dependencies for precise map element detection. VectorMapNet [22] first employs a two-stage framework to decode the vectorized representation of map elements. The method first uses a DETR-like Transformer [110] to extract key points from BEV features, followed by an autoregressive Transformer that generates the vertex sequence

for each map element. To address the efficiency bottleneck of autoregressive decoding in [22], MapTR [56] (Figure 7a) employs a single-stage decoder for parallel decoding. It initializes hierarchical queries with instance-level and point-level embeddings. It uses a DETR-like Transformer to enhance their interaction, producing an ordered point sequence for the detected elements. The follow-up work [76] introduces a one-to-many matching mechanism to speed up convergence and employs auxiliary dense supervision to enhance the performance of map element detection. MapVR [62] introduces rasterization to help the model capture intricate map details. It features a differentiable rasterizer that provides fine-grained geometric shape supervision for vectorized outputs. MapDistill [71] introduces a three-level cross-modal knowledge distillation framework that transfers knowledge from a camera–LiDAR fusion model to a camera model using a teacher–student approach.



(**a**)



(**b**)

Figure 7. Comparison of the pipelines of two Transformer-based map element detection methods. (**a**) MapTR [56] uses a single-stage DETR-like Transformer [110] for parallel decoding of ordered point sequences for map elements. (**b**) MGMap [67] uses instance masks to enhance element queries for precise localization and uses mask patches to refine point position predictions.

Several methods improve the query and decoder designs in Transformer to enhance the accuracy of map element detection. HIMap [66] introduces HIQuery, a hybrid representation that integrates point-level and element-level information for map elements. Its hybrid decoder employs a point-element interaction module that iteratively fuses these information types, enabling accurate prediction of point coordinates and element shapes. In contrast, MapQR [75] proposes a "scattering and aggregation" mechanism to enhance instance queries. This mechanism distributes each query into sub-queries with distinct positional embeddings to gather features from different locations, which are then recombined into a unified instance query for enriched map element representations. MGMap [67], as shown in Figure 7b, leverages learned masks to enhance the detection of map elements. The method integrates global structural information from instance masks for accurate localization. It then extracts local semantic features from mask patches around predicted points to adjust point positions. Additionally, InsMapper [68] leverages point-wise correlations within map element instances to improve detection accuracy. The method enables information sharing and strengthens feature associations within each instance, resulting in smoother and more coherent map element detection in complex scenes. ADMap [74] reduces jitter in vectorized point sequences during map element detection.

Other methods improve the vectorized representation of map elements to enhance detection accuracy and efficiency. BeMapNet [57] employs piecewise Bezier curves to capture the shapes of complex map elements. The representation decomposes a curve into low-degree Bezier segments, each efficiently capturing local geometry with fewer control points to represent diverse map structures. PivotNet [60] introduces a pivot-based vectorized representation that selects key geometric points on map elements to create a compact and precise vectorized map. Additionally, ScalableMap [61] proposes a hierarchical sparse map representation that samples map elements at varying densities to balance computational cost and accuracy. DPFormer [64] introduces a compact Douglas–Peucker point representation. It selects key points based on curvature, increasing point density in curved sections while reducing points in straight sections, effectively minimizing redundancy and preserving map element structure. GeMap [69] proposes G-Representation. This approach uses displacement vectors between adjacent points to describe local geometric features of map instances, ensuring invariance to rotation and translation.

Several methods leverage short-term temporal information to improve map element detection's accuracy and temporal consistency. StreamMapNet [63], as shown in Figure 8a, introduces a streaming temporal fusion mechanism that employs two strategies. Query propagation retains high-confidence element queries to the next frame, while BEV fusion aligns and fuses BEV features from consecutive frames. Building on this, SQD-MapNet [72] proposes a stream query denoising mechanism to improve the temporal consistency. The mechanism adds random noise to previous frame elements and then recovers the geometric shapes for the current frame through denoising, enhancing the model's ability to capture temporal changes. In contrast, MapTracker [73] adopts a stacking strategy and introduces a dual-memory mechanism. The BEV memory module selects BEV features from historical frames by geometric distance and fuses them with current frame information. In contrast, the vector memory module filters historical map element queries by distance and refines current frame elements via per-instance cross-attention. PrevPredMap [79] integrates high-level information from previous predictions—such as map element categories, confidence, and location—into current frame predictions, improving the quality of map element detection.

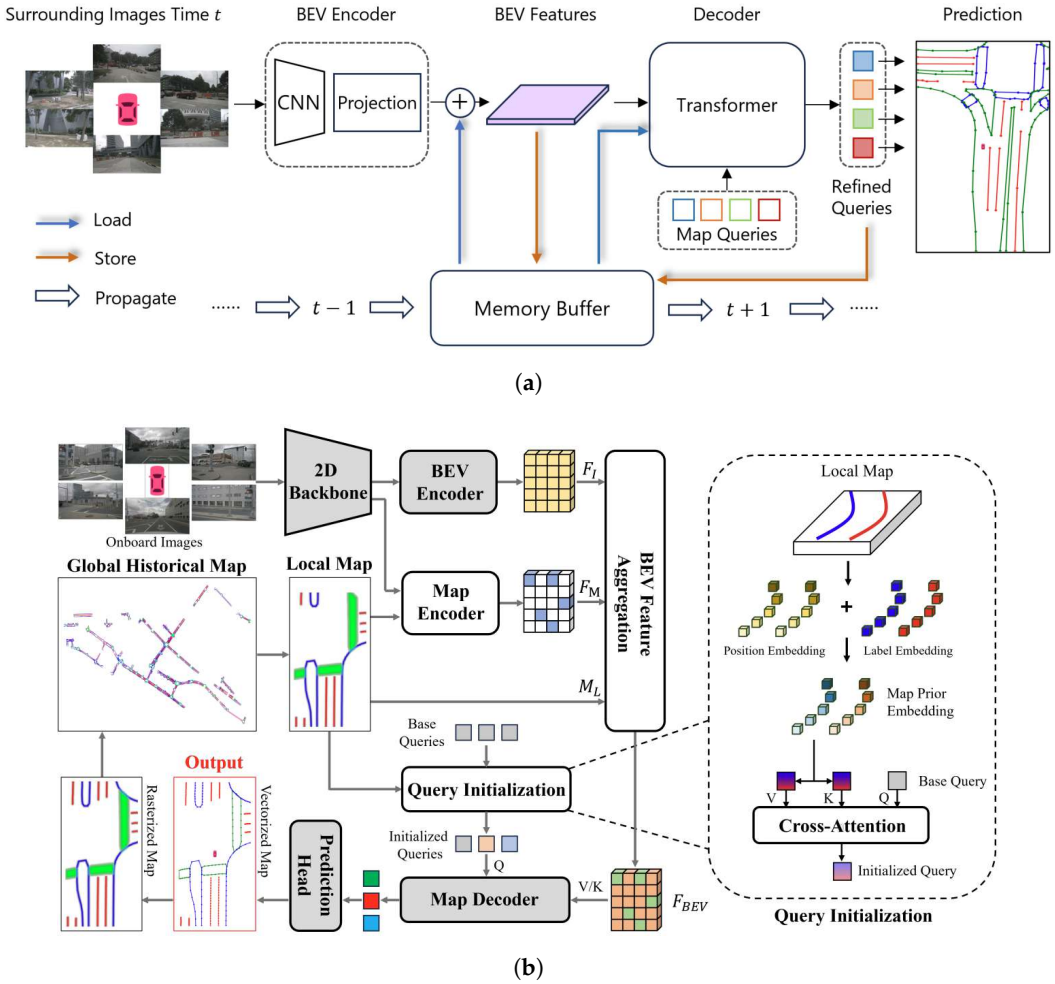


Figure 8. Comparison of temporal fusion (short-term and long-term) in two Transformer-based map element detection methods. (**a**) StreamMapNet [63] aligns and fuses BEV features from consecutive frames and propagates high-confidence element queries to the next frame. (**b**) HRMapNet [70] fuses BEV features with rasterized map features to enrich information and rasterizes vectorized map predictions to maintain a global historical map.

Another direction is to leverage long-term temporal information derived from historical maps to enhance map element detection. NMP [58] introduces the concept of "neural map prior". The method employs cross-attention to integrate BEV features with global prior features, generating enhanced BEV features for improved detection. It then utilizes a gated recurrent unit (GRU) to dynamically update the global "neural map prior" with the enhanced BEV features. Similarly, HRMapNet [70] (Figure 8b) adopts a strategy that utilizes historical rasterized maps. The method designs a feature aggregation module to fuse BEV features with rasterized map features, enriching for improved map element detection. It then rasterizes the vectorized map predictions and maps them to the global map to facilitate continuous updates. Additionally, DTCLMapper [77] introduces a dual-temporal consistent module that leverages short- and long-term information. Using contrastive learning enhances temporal consistency by aligning same-category features across frames while distinguishing different ones. It also projects vectorized map predictions onto a global rasterized map, using occupancy constraints to ensure spatial consistency across frames. P-MapNet [78] introduces two map prior modules to improve distant map element detection. SDMap provides road skeleton information integrated with BEV features via cross-attention, while HDMap uses a masked autoencoder to learn HD map distribution patterns, refining predictions and correcting structures.

4.2.3. Discussion on Map Element Detection Methods

Map element detection methods can be classified into two types based on MD techniques, each with unique advantages and limitations. First, CNN-based methods utilize CNN to process the image or BEV features hierarchically, producing dense outputs transformed into vectorized representations of map elements via post-processing or association. While effectively extracting local features and reducing model complexity via weight sharing, they struggle with long-range dependencies, making them less suitable for map elements with long spatial spans like lanes and sidewalks. Second, Transformer-based methods utilize self-attention to capture relationships between patches in image or BEV features, enabling direct generation of vectorized representations through parallel or autoregressive decoding. This approach enhances understanding of long-range dependencies in geometric structures but also increases model complexity and computational demands, requiring significant training data for effective generalization in unseen road environments.

Table 4 shows the results of map element detection methods on the nuScenes [29] validation set, revealing two key observations. First, Transformer-based methods significantly outperform CNN-based methods. In the multi-camera setup, the worst Transformer-based method, InsMapper [68], has an mAP 11.6% higher than the best CNN-based method, InstaGraM [59]. In the multi-camera and LiDAR setup, InsMapper's mAP exceeds that of HDMapNet [21] by 30.0%. This superiority is likely due to Transformers' self-attention mechanism, which effectively captures long-range dependencies and global context, enhancing their ability to detect map elements that span large distances. Second, integrating LiDAR input modality significantly improves vision-based methods. For example, HDMapNet [21] achieves an 8.0% increase in mAP, ADMap [74] improves by 7.9%, and MGMap [67] rises by 6.9%. This enhancement results from Supplementary Information provided by LiDAR point clouds, including (1) 3D geometric information, such as vehicle points in lanes that offer context for map element detection, and (2) reflectance intensity, where the high reflectance of lane markings sharply contrasts with the low reflectance of rough road surfaces.

Table 4. Performance comparison of map element detection methods on the nuScenes [29] validation set. Under **Backbone**, "EB0" is for EfficientNet-B0 [111], "R50" is for ResNet-50 [112], "PP" is PointPillars [113], and "Sec" is SECOND [114]. Under **Modality**, "MC" is for multi-camera and "L" is for LiDAR. Under **MD Type**, "CNN" and "Transformer" denote two types of map element detection methods based on map decoding techniques. The symbol * indicates results from original papers, and † denotes results from [56].

Method	Backbone	Epochs	Modality	MD Type	$AP_{ped.}$	$AP_{div.}$	$AP_{bou.}$	mAP
HDMapNet † [21]	EB0	30	MC	CNN	14.4	21.7	33.0	23.0
InstaGraM * [59]	EB0	30	MC	CNN	40.8	30.0	39.2	36.7
InsMapper * [68]	R50	24	MC	Transformer	44.4	53.4	52.8	48.3
MapTR * [56]	R50	24	MC	Transformer	46.3	51.5	53.1	50.3
MapVR * [62]	R50	24	MC	Transformer	47.7	54.4	51.4	51.2
PivotNet * [60]	R50	30	MC	Transformer	53.8	55.8	59.6	57.4
BeMapNet * [57]	R50	30	MC	Transformer	57.7	62.3	59.4	59.8
MapTRv2 * [76]	R50	24	MC	Transformer	59.8	62.4	62.4	61.5
ADMap * [74]	R50	24	MC	Transformer	63.5	61.9	63.3	62.9
SQD-MapNet * [72]	R50	24	MC	Transformer	63.0	62.5	63.3	63.9
MGMap * [67]	R50	30	MC	Transformer	61.8	65.0	67.5	64.8
MapQR * [75]	R50	30	MC	Transformer	63.4	68.0	67.7	66.4
HIMap * [66]	R50	30	MC	Transformer	62.6	68.4	69.1	66.7
HDMapNet † [21]	EB0 & PP	30	MC & L	CNN	16.3	29.6	46.7	31.0
InsMapper * [68]	R50 & Sec	24	MC & L	Transformer	56.0	63.4	71.6	61.0
MapVR * [62]	R50 & Sec	24	MC & L	Transformer	60.4	62.7	67.2	63.5
MapTRv2 * [76]	R50 & Sec	24	MC & L	Transformer	65.6	66.5	74.8	69.0
ADMap * [74]	R50 & Sec	24	MC & L	Transformer	69.0	68.0	75.2	70.8
MGMap * [67]	R50 & Sec	24	MC & L	Transformer	67.7	71.1	76.2	71.7

4.3. Lane Graph Construction for Vectorized Maps

Lane graph construction algorithms dynamically generate vectorized maps that depict the road environment around the ego vehicle as directed lane graphs, enriched with detailed geometric and topological information. A key challenge in this process is reasoning about the topological relationships between lanes and traffic elements within complex road environments. This encompasses the connectivity and adjacency of lanes (such as intersections, forks, and merges) and the correspondence between traffic elements and lanes. To tackle this challenge, leading lane graph construction algorithms propose various topology reasoning (TR) methods to effectively identify and analyze the intricate relationships within road scene structures, enhancing the accuracy and practicality of the generated lane graphs.

Recent research [19,36,84,85,94] has focused on enhancing this TR module, leading us to categorize lane graph construction methods into two main types based on TR techniques. The first category, "**Single-step-based Methods**", completes the TR of the entire lane graph in a single step using information from the driving scene. The second category, "**Iteration-based Methods**", conducts TR over iterative steps, where each step analyzes, reasons, and adjusts specific sections of the lane graph, resulting in a more refined topological structure.

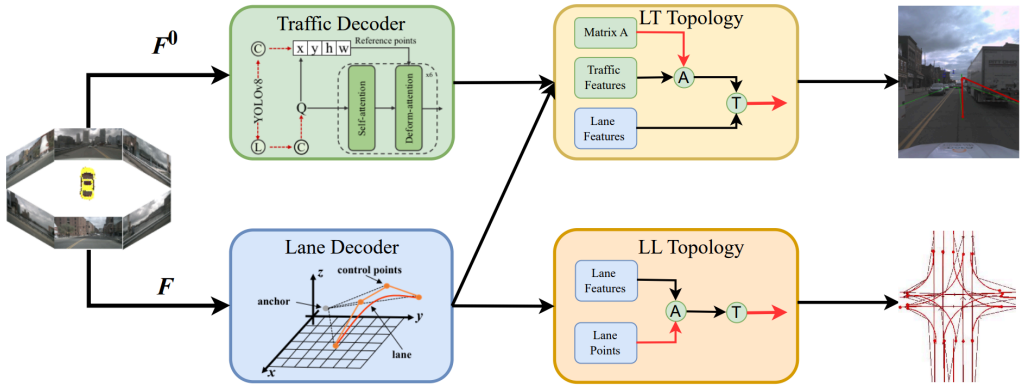
4.3.1. Single-Step-Based TR for Lane Graph Construction

Single-step-based methods complete the TR of the lane graph in a single step, deducing the relationships between all lanes and traffic elements in the driving scene. These methods trace back to STSU [19], which infers lane connectivity in a lane graph in a single step using an association head. It first employs a DETR-like Transformer [110] to process feature maps and extract lane queries. Then, MLP-based task heads decode the geometric and topological information from these queries, outputting lane existence probabilities, Bézier curve control points, and connection probabilities between lane pairs to construct a directed lane graph. LaneSegNet [86] adopts a similar TR approach to generate lane graphs from lane segment representations, simultaneously providing geometric, semantic, and topological information. In contrast, LaneGAP [92] adopts a post-processing strategy to convert individual lane predictions into a complete lane graph. This method discretizes lane predictions into vertices and merges nearby vertices to restore the topological structures of lane forks and merges.

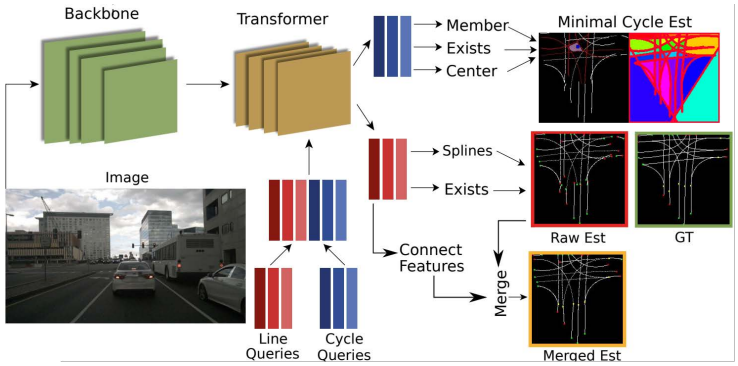
High-quality detection is crucial for the TR of lane graphs, with some methods enhancing lane and traffic element detectors to achieve this. TopoMLP [87], as shown in Figure 9a, presents an efficient lane graph construction pipeline. It first uses a PETR-like Transformer [115] to detect lanes and then integrates YOLOv8 as an auxiliary detector to enhance small traffic element detection. Finally, two MLP heads infer two types of topology relationships to generate a complete lane graph. Some methods [81,89] employ temporal aggregation with multi-frame information to address occlusion, enhancing the quality of generated lane graphs. LaneMapNet [89] introduces a curve region-aware attention mechanism, which learns curve-shape features to improve lane regression. Additionally, RoadPainter [91] introduces a point-mask optimization mechanism that generates instance masks from initial lane predictions, samples representative points, and fuses them with the original lane points to refine predictions.

Other methods enhance the topological accuracy of lane graphs by incorporating auxiliary supervision signals based on prior knowledge. TPLR [20] (Figure 9b) introduces minimal cycles in the lane graph—the smallest closed curves formed by lane intersections—to accurately capture its topological structure. The method employs a Transformer to process lane and cycle queries simultaneously, followed by an MLP to output the cover of minimal cycles, helping the model learn the correct order of lane intersections. Similarly, ObjectLane [83] models the relationship between traffic objects and lanes as a clustering problem, with lanes as cluster centers and traffic objects as data points. It uses a Trans-

former to generate association information, assigning each detected object a probability distribution for its most likely corresponding lane. This approach ensures that the lane graph reflects road geometry and traffic participants' distribution. LaneWAE [82] leverages the dataset's prior distribution to enhance lane graph predictions. It uses a Transformer-based Wasserstein autoencoder to capture a latent space representation of lane structures. The method then refines initial predictions by optimizing latent space vectors to align with the learned prior.



(**a**)



(**b**)

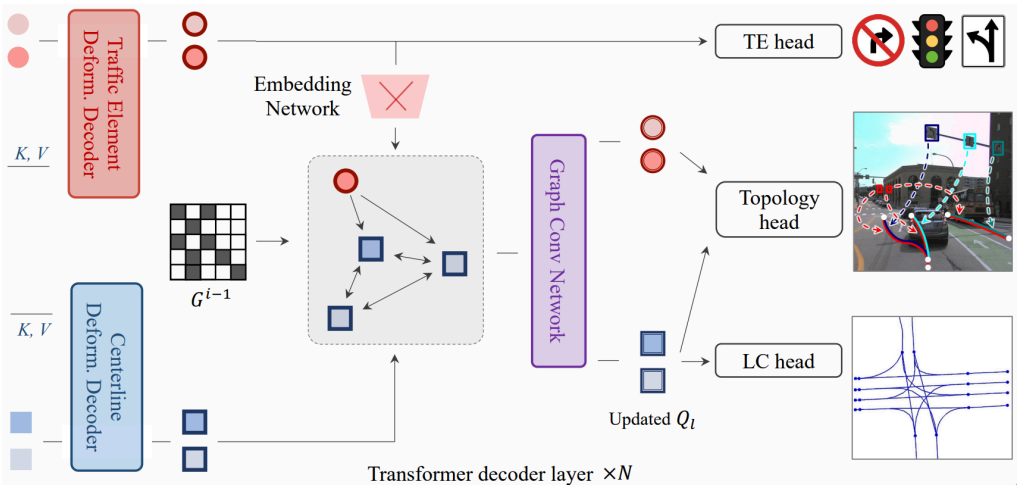
Figure 9. Comparison of the pipelines for two single-step-based lane graph construction methods. (**a**) TopoMLP [87] uses two Transformers for lane and traffic element queries, followed by MLPs to predict the topological relationships between paired queries. (**b**) TPLR [20] uses a Transformer to process lane and minimal cycle queries simultaneously, followed by joint decoding of the lane graph and the cover of minimal cycles.

4.3.2. Iteration-Based TR for Lane Graph Construction

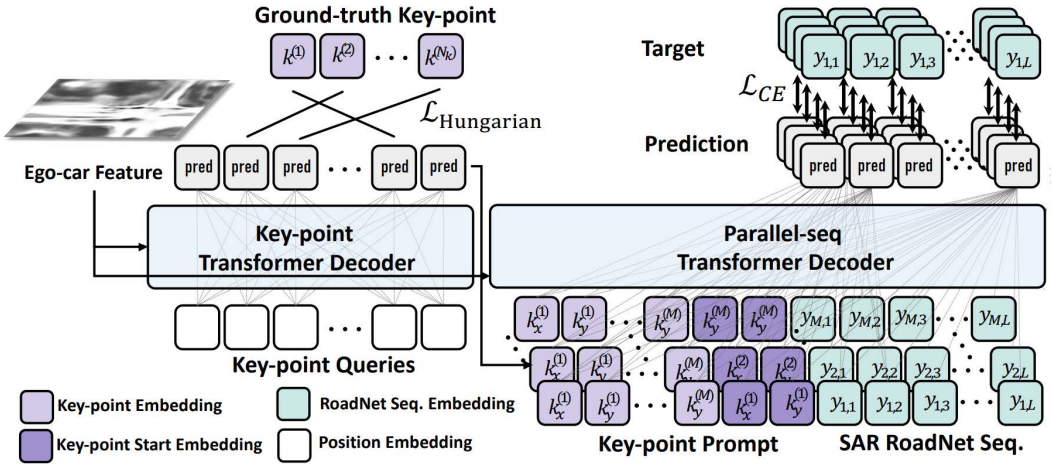
Iteration-based methods conduct the TR of the lane graph in iterative steps, where each step analyzes, reasons, and adjusts specific relationships between lanes and traffic elements, gradually refining the overall structure. TopoNet [85], as depicted in Figure 10a, first leverages a scene GNN for the iterative refinement of lane graph topologies. It begins with a DETR-like Transformer [110] to extract queries for lane and traffic elements, forming two initial graphs. A graph convolutional network (GCN) then performs iterative message passing and feature updating, refining the queries with spatial and semantic context from neighboring nodes, ultimately creating a comprehensive lane graph. Similarly, TopoLogic [94] employs an iterative TR pipeline that integrates two lane topologies to enhance reasoning in complex driving scenes. The method first calculates the geometric distances between predicted lanes to assess their connectivity, and then evaluates the similarity of lane queries in semantic space to address geometric reasoning limitations. Finally, it fuses the adjacency matrices from both topologies to create a more accurate lane

graph. Additionally, CGNet [90] combines GCN with GRU to iteratively optimize the lane graph. The GRU's memory mechanism retains information from previous layers, allowing TR to leverage both current data and accumulated knowledge. SMERF [88] combines road topology from SD maps with vehicle sensor data to enhance lane topology reasoning.

Recently, researchers have developed various mask-based mechanisms to improve lane detection. One significant approach is TopoMaskV2 [93], which employs a masked attention mechanism. The method generates Bezier control points and mask embeddings from lane queries, then focuses on the masked regions to update these queries effectively. It integrates the outputs from the mask head and Bezier head to achieve smoother and more accurate lane predictions.



(**a**)



(**b**)

Figure 10. Comparison of the TR module in two iteration-based lane graph construction methods. (**a**) TopoNet [85] uses two Transformers for lane and traffic element queries, followed by a GCN for iterative message passing and feature updating. (**b**) RoadNetTransformer [84] (semi-autoregressive) first predicts lane key points in parallel and then autoregressively generates local sequences for lane graphs.

Some methods perform local TR in a sequential manner to construct a complete lane graph. CenterLineDet [80] mimics expert annotators to construct a lane graph vertex by vertex. It utilizes the VT network to generate a BEV heatmap and extract the initial vertex set, while a DETR-like Transformer [110] then iteratively predicts the next vertex position based on the ROI heatmap. Finally, the local lane graphs are combined to create a comprehensive global lane graph. In contrast, RoadNetTransformer [84] (Figure 10b) introduces the

RoadNet Sequence representation, effectively capturing the topological structure of the lane graph for sequential TR. It features three Transformer-based decoder architectures with distinct strategies: the semi-autoregressive version predicts lane key points in parallel and autoregressively generates the local sequences, while the non-autoregressive version performs full parallel predictions on the masked sequence and iteratively refines low-confidence tokens to enhance sequence quality. Similarly, LaneGraph2Seq [36] introduces a graph sequence representation for lane graphs. It employs an autoregressive Transformer to sequentially generate the vertex and edge sequences, ultimately reconstructing them into the complete lane graph structure.

4.3.3. Discussion on Lane Graph Construction Methods

Lane graph construction methods can be categorized into two types based on TR techniques, each with distinct advantages and limitations. First, single-step-based methods perform the TR of the entire lane graph in one step using driving scene information. They are easy to implement and have short processing times, but one-step reasoning lacks flexibility, often leading to suboptimal performance in complex traffic scenarios. For example, in large intersections with intricate topological structures, a single-step approach may struggle to accurately capture the relationships among multiple lanes, turn lanes, and traffic signals. Second, iteration-based methods perform TR by reasoning and refining specific parts of the lane graph through multiple steps. While this approach allows for fine-tuning predictions and offers greater flexibility, it also incurs higher computational costs and longer processing times, making it challenging to respond effectively to sudden obstacles or changing traffic patterns. Therefore, a key research direction is to balance the accuracy and speed of TR to meet real-time demands while ensuring high-quality lane graphs.

Tables 5 and 6 present the results of lane graph construction methods on the nuScenes [29] validation set and the OpenLane-V2 [31] dataset, revealing two key observations. First, iteration-based methods significantly outperform single-step-based methods. For example, LaneGraph2Seq [36] achieves the highest C-F score of 63.2 in Table 5, while TopoLogic [94] and TopoMaskV2 [93] secure the best OLS scores in Table 6—41.6 and 41.7 for subset A, and 39.6 and 43.9 for subset B, respectively. This superiority likely arises from their ability to iteratively analyze and refine lane graphs, enhancing TR in complex traffic scenes. Second, incorporating the SD map as an additional data source markedly improves the model's TR ability. In Table 6, all three methods show significant OLS score increases after integrating the SD map—SMERF [88] (TopoNet) by 3.8, RoadPainter [91] by 3.7, and TopoLogic [94] by 3.5. This improvement stems from the SD map's provision of crucial topological information, enhancing the model's understanding of global road layouts and boosting accuracy in distant and occluded areas.

Table 5. Performance comparison of lane graph construction methods on the nuScenes [29] validation set (PON [38] split). Under **TR Type**, "Single-step" and "Iteration" denote two types of lane graph construction methods based on topology reasoning techniques. The symbol * indicates results from original papers, and "–" denotes that the statistic is unavailable.

Method	TR Type	M-Prec	M-Recall	M-F-score	Detect	C-Prec	C-Recall	C-F-score
STSU * [19]	Single-step	60.7	54.7	57.5	60.6	60.5	52.2	56.0
TPLR * [20]	Single-step	–	–	58.2	60.2	–	–	55.3
ObjectLane * [83]	Single-step	–	–	64.2	70.6	–	–	57.2
VideoLane * [81]	Single-step	–	–	59.0	60.3	–	–	61.7
LaneWAE * [82]	Single-step	–	–	57.0	61.2	–	–	62.9
LaneMapNet * [89]	Single-step	71.5	64.8	67.9	–	63.2	62.9	63.0
LaneGraph2Seq * [36]	Iteration	64.6	63.7	64.1	64.5	69.4	58.0	63.2

Table 6. Performance comparison of lane graph construction methods on the OpenLane-V2 [31] dataset (v1.0 metrics). Under **TR Type**, "Single-step" and "Iteration" denote two types of lane graph construction methods based on topology reasoning techniques. **SD Map** indicates the use of a standard-definition map. The symbol * indicates results from original papers, and † denotes results reproduced by [85].

Data	Method	TR Type	SD Map	DET_l	DET_t	TOP_{ll}	TOP_{lt}	OLS
subset_A	STSU † [19]	Single-step	✗	12.7	43.0	0.5	15.1	25.4
	TopoNet * [85]	Iteration	✗	28.5	48.1	4.1	20.8	35.6
	TopoMLP * [87]	Single-step	✗	28.3	50.0	7.2	22.8	38.2
	RoadPainter * [91]	Single-step	✗	30.7	47.7	7.9	24.3	38.9
	TopoLogic * [94]	Iteration	✗	29.9	47.2	18.6	21.5	41.6
	TopoMaskV2 * [93]	Iteration	✗	34.5	53.8	10.8	20.7	41.7
	SMERF * [88] (TopoNet)	Iteration	✓	33.4	48.6	7.5	23.4	39.4
	RoadPainter * [91]	Single-step	✓	36.9	47.1	12.7	25.8	42.6
	TopoLogic * [94]	Iteration	✓	34.4	48.3	23.4	24.4	45.1
subset_B	STSU† [19]	Single-step	✗	8.2	43.9	0.0	9.4	21.2
	TopoNet * [85]	Iteration	✗	24.3	55.0	2.5	14.2	33.2
	TopoMLP * [87]	Single-step	✗	26.6	58.3	7.6	17.8	38.7
	RoadPainter * [91]	Single-step	✗	28.7	54.8	8.5	17.2	38.5
	TopoLogic * [94]	Iteration	✗	25.9	54.7	15.1	15.1	39.6
	TopoMaskV2 * [93]	Iteration	✗	41.6	61.1	12.4	14.2	43.9

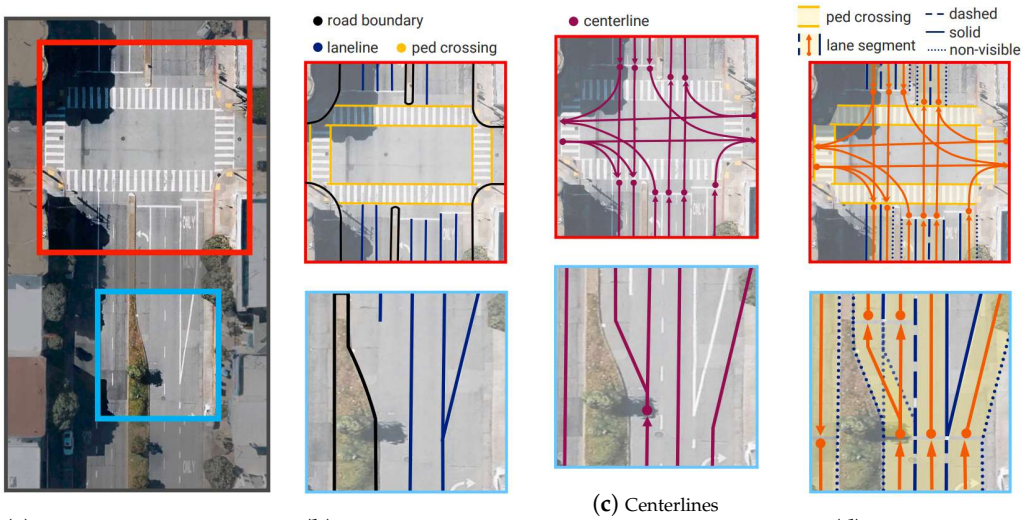
5. Challenges and Future Trends

This section presents the challenges and future trends in online HD map construction, focusing on standardized representation design, multi-task learning, and multi-modality fusion.

5.1. Standardized Representation Design

A challenge in online HD map construction is the lack of a standardized representation. The three main representations, each with its strengths and limitations are the following: (1) map pixel grids in map segmentation, which capture fine-grained details but cannot distinguish between instances like lanes and crosswalks and are costly in storage and computation; (2) vectorized lane lines in map element detection, which efficiently represent road geometry but lack topological information, such as lane relationships at intersections; and (3) centerlines and traffic elements in lane graph construction, which effectively capture road network topology but lack detailed lane features like type, direction, and width. Therefore, representation design is crucial for online HD map construction, affecting the description of the surrounding road environment and the transmission of structured information to the decision-making module.

An important research direction is to develop a unified map representation that combines the strengths of existing methods while addressing their limitations. LaneSegNet [86], as depicted in Figure 11, introduces lane segment representation, integrating vectorized lane lines, centerlines, and road attributes to capture geometric, semantic, and topological information. Specifically, it includes (1) geometric information, such as lane centerlines and boundaries, with offsets defining boundary positions and polygons delineating drivable areas; (2) semantic information, including lane types (e.g., traffic lanes, pedestrian crossings) and boundary line types (e.g., solid, dashed, invisible) to encode lane characteristics and traversability; (3) topological information, represented as a lane graph with nodes for lane segments and edges for connectivity, stored in an adjacency matrix to model lane relationships like merging and diverging.

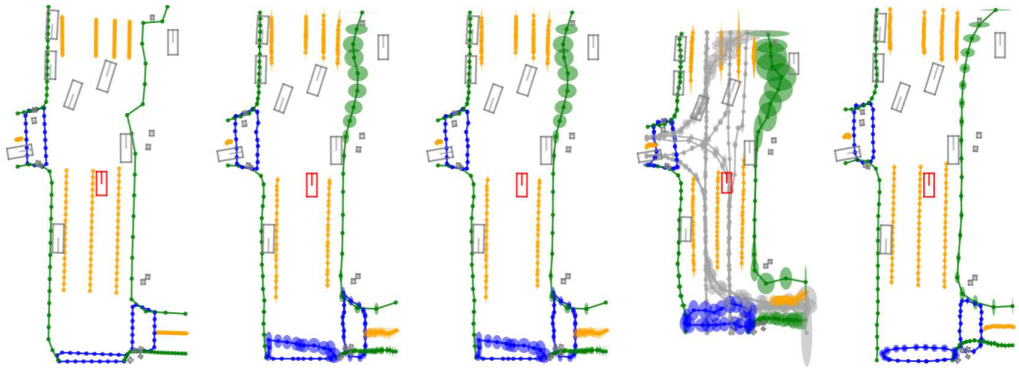


(**a**) BEV of the region of interest (**b**) Vectorized lane lines (**c**) Centerlines and topology (**d**) Lane segment

Figure 11. Comparison of lane segment representation [86] with two alternative map representations.

Another research direction is enhancing map representations by adding 3D structures (e.g., interchanges, elevated roads) and dynamic features (e.g., traffic lights, road signs) to better capture complex road environments. For example, OpenLane-V2 [31] expands the centerline lane graph with 13 types of traffic elements, including traffic lights, directional signs, restrictive signs, and special maneuver signs, while defining their relationships with lanes. Future research could analyze the diverse and heterogeneous elements within traffic scenes and their intricate interactions to provide more detailed structured information for scene understanding.

The third research direction is designing map representations that integrate seamlessly with downstream modules. Gu et al. [116] (Figure 12) propose an uncertainty-based map representation that uses probabilistic modeling to output map elements' positions, categories, and uncertainties. This approach significantly enhances its utility in trajectory prediction tasks. Specifically, positional uncertainty is modeled with a Laplace distribution to capture prediction errors, while categorical uncertainty reflects confidence levels via probability distributions. These uncertainties enable trajectory prediction models to dynamically adjust the weighting of map elements, improving accuracy, robustness, and training efficiency.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 12. Comparison of uncertainty-based map representations [116] integrated into various online HD map construction methods. (**a**) Ground truth. (**b**) MapTR [56]. (**c**) MapTRv2 [76]. (**d**) MapTRv2-CL [76]. (**e**) StreamMapNet [63].

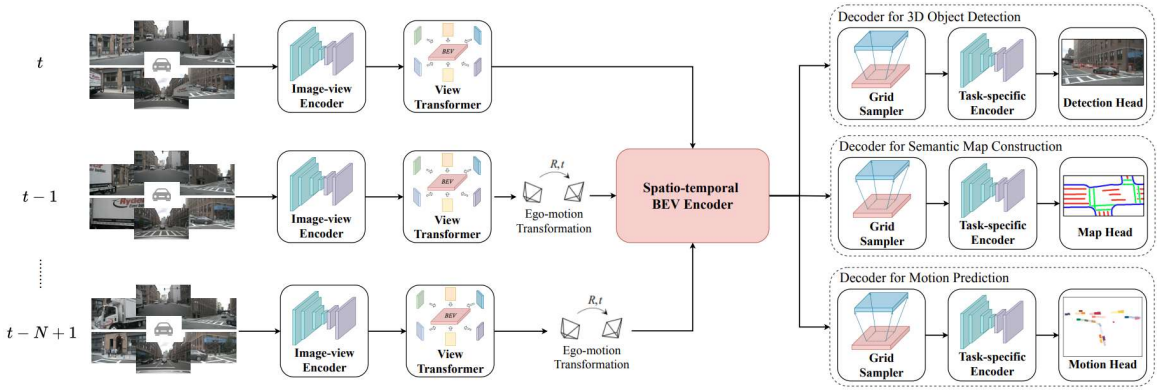
5.2. Multi-Task Learning

Multi-task learning (MTL) trains a single model to perform multiple related tasks simultaneously, providing distinct advantages over separate task-specific models. First, MTL reduces computational and storage demands by sharing model structures and parameters, reducing the need for multiple models and lowering inference costs. Second, MTL improves generalization by capturing complementary information across tasks, which helps prevent overfitting and enhances performance on new data [117]. Consequently, MTL is well-suited for online HD map construction, as it can effectively integrate complementary information from related tasks to create more comprehensive local HD maps.

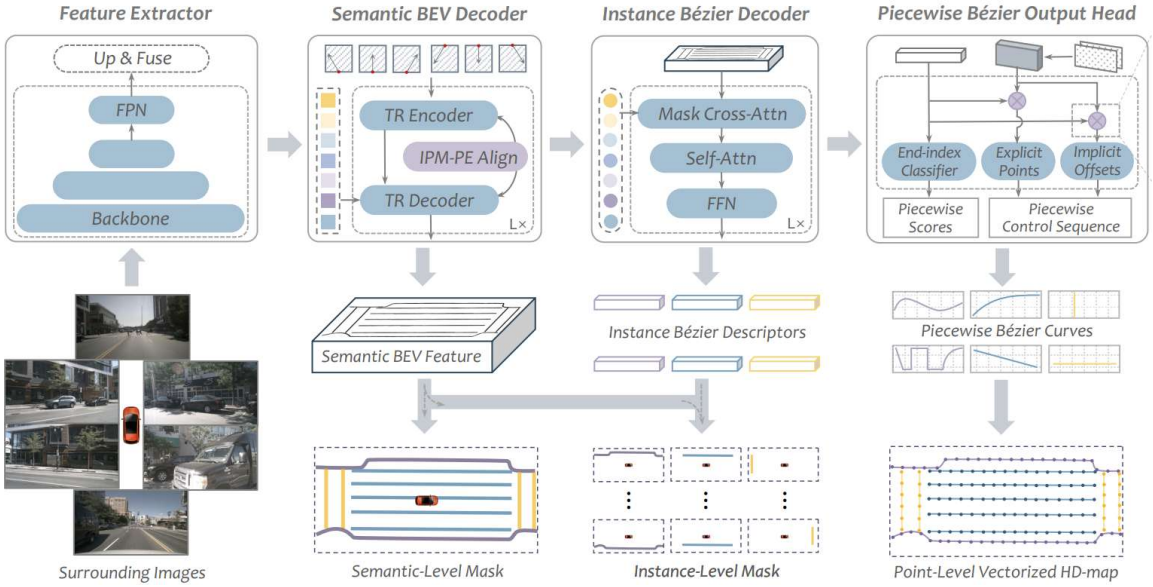
Some methods integrate other perception and prediction tasks to enhance online HD map construction. These tasks include (1) semantic segmentation [73–76], which classifies different regions in an image to provide insights into their categories and boundaries, helping the model understand the scene's semantic structure; (2) depth estimation [73–76], which infers the distance between image pixels and the camera, offering information on the spatial location and geometric shapes of objects for accurate 3D modeling; (3) 3D object detection [19,45,46,49,50,52,53,55], which identifies and locates traffic-related objects, providing details on their categories, sizes, and relative positions, thus contributing contextual information; and (4) motion prediction [50], which estimates future trajectories of objects, enhancing the model's understanding of complex and dynamic traffic scenarios.

Another direction is integrating multiple map-related tasks to enhance online HD map construction. BeMapNet [57] (as depicted in Figure 13b) and PivotNet [60] introduce map segmentation and instance segmentation of map elements, addressing sparse supervision in map element detection. HIMap [66] adopts a similar MTL strategy, while MGMap [67] and MapTracker [73] each focus on one of the tasks. MapTRv2 [76] enhances map element detection by incorporating map segmentation, semantic segmentation, and depth estimation from a perspective view. Later works [69,74,75] follow similar approaches. To combine map element detection with lane graph construction, LaneSegNet [86] introduces lane segment representation that captures both geometric boundaries and topological relationships. TopoMaskV2 [93] enhances lane detection with instance segmentation. SemVecNet [118] integrates map segmentation to generate an intermediate representation, enhancing adaptability to different sensor platforms and preventing overfitting to specific sensor configurations. Mask2 Map [119] also uses BEV segmentation masks to refine map element detection. It introduces a positional query generator and geometric feature extractor to extract local contextual information within the masks, enhancing the query features to improve detection accuracy.

However, integrating perception and prediction tasks for multi-task learning in online HD map construction does not always improve performance and can even result in "negative transfer". For instance, as shown in Table 7, M^{2}BEV [49], BEVFormer [45], and PETRv2 [55] experience a decrease in map segmentation mIoU by 1.7%, 2.2%, and 2.3%, respectively, after integrating 3D object detection, while Ego3RT [46] sees a 9.3% increase. Similarly, BEVerse [50] shows a 4.5% drop in mIoU after integrating both 3D object detection and motion prediction. In contrast, Table 8 demonstrates that MapTRv2 [76] achieves increases of 5.1%, 6.4%, and 7.0% in map element prediction mAP by progressively integrating depth estimation, map segmentation, and semantic segmentation. Future research could explore the relationships between online HD map construction and other tasks to drive performance improvements.



(**a**)



(**b**)

Figure 13. Comparison of the MTL pipeline in two online HD map construction methods. (**a**) BEVerse [50] presents a unified framework for map segmentation, 3D object detection, and motion prediction. (**b**) BeMapNet [57] presents a unified framework for map segmentation, map element detection, and instance segmentation.

Table 7. Performance comparison of MTL map segmentation methods on the nuScenes [29] validation set. Under **Task Head**, "MapSeg" is map segmentation, "ObjDet" is 3D object detection, and "MotPre" is motion prediction. The symbol * indicates results from [43], † denotes results from [24], and "–" denotes that the statistic is unavailable.

Method	Task Head			Map Segmentation							
	MapSeg	ObjDet	MotPre	Drivable	Lane	Ped. Cross.	Walkway	Carpark	Divider	Boundary	mIoU
M^{2}BEV * [49]	✓	✗	✗	77.2	40.5	–	–	–	–	–	58.9
M^{2}BEV * [49]	✓	✓	✗	75.9	38.0	–	–	–	–	–	57.0
BEVFormer * [45]	✓	✗	✗	80.1	25.7	–	–	–	–	–	52.9
BEVFormer * [45]	✓	✓	✗	77.5	23.9	–	–	–	–	–	50.7
PETRv2 † [55]	✓	✗	✗	80.5	47.4	–	–	–	–	–	64.0
PETRv2 † [55]	✓	✓	✗	79.1	44.3	–	–	–	–	–	61.7
Ego3RT * [46]	✓	✗	✗	74.6	–	33.0	42.6	44.1	36.6	–	46.2
Ego3RT * [46]	✓	✓	✗	79.6	–	48.3	52.0	50.3	47.5	–	55.5
BEVerse * [50]	✓	✗	✗	–	–	44.9	–	–	56.1	58.7	53.2
BEVerse * [50]	✓	✓	✓	–	–	39.0	–	–	53.2	53.9	48.7

Table 8. Performance comparison of MTL MapTRv2 [76] on the nuScenes [29] validation set. Under **Task Head**, "MapEle" is map element detection, "DepEst" is for depth estimation, "MapSeg" is map segmentation, and "SemSeg" is for semantic segmentation. The symbol * indicates results from the original paper.

Method	Task Head				Map Element Detection			
	MapEle	DepEst	MapSeg	SemSeg	$AP_{ped.}$	$AP_{div.}$	$AP_{bou.}$	mAP
MapTRv2 * [76]	✓	✗	✗	✗	44.9	51.9	53.5	50.1
MapTRv2 * [76]	✓	✓	✗	✗	49.6	56.6	59.4	55.2
MapTRv2 * [76]	✓	✓	✓	✗	52.1	57.6	59.9	56.5
MapTRv2 * [76]	✓	✓	✓	✓	53.2	58.1	60.0	57.1

5.3. Multi-Modality Fusion

Multi-modality fusion (MMF) combines data from different modalities—such as images, point clouds, and SD maps—for processing and analysis. Each AV sensor has unique strengths and limitations [120]: (1) cameras capture rich visual details like color and texture but are sensitive to lighting, weather, and lack depth; (2) LiDAR provides precise depth data and functions in day or night but lacks color/texture and can be affected by extreme weather; and (3) radar is robust to lighting and harsh weather, detects object velocity, but has lower resolution and cannot capture precise shapes. In addition, SD maps offer basic road structures with broad coverage and low cost but lack high-precision traffic details, such as lane markings, traffic signs, and signals. Consequently, MMF is well-suited for online HD map construction, as it can effectively integrate complementary information from multi-modal data to create more accurate local HD maps.

Some methods integrate data from multiple sensors to enhance online HD map construction. BEVFusion [52], as depicted in Figure 14a, pioneers the fusion of camera and LiDAR data in the BEV space. The method projects camera and LiDAR features into a unified BEV space, concatenates them, and uses a convolutional BEV encoder to resolve local misalignment. Subsequent studies [21,67,74,76,78] adopt a similar middle fusion strategy. Simple-BEV [53] enhances performance by fusing camera and radar data, achieving near-LiDAR accuracy without relying on LiDAR. Additionally, SuperFusion [65] introduces a multi-level fusion technique to combine LiDAR and camera inputs, enabling long-range map element detection at distances up to 90 meters.

Another direction is to integrate SD maps and other prior maps to enhance online HD map construction. SMERF [88] uses road topology from SD maps to improve lane graph construction. The method first encodes polyline sequences from SD maps, then extracts road topology using a Transformer encoder, and finally fuses SD map features with BEV features through multi-head cross-attention. Subsequent works [91,94] adopt a similar strategy for integrating SD map data. In contrast, P-MapNet [78] rasterizes SD maps, encodes them with CNNs, and adaptively fuses SD map features with BEV features. NMP [58], as shown in Figure 14b, constructs global neural map priors from BEV features of previous traversals, enhancing local map inference. Similarly, HRMapNet [70] uses historical rasterized maps to complement online perception data. DTCLMapper [77] maintains a grid map and uses occupancy states to ensure the geometric consistency of map elements.

As shown in Table 9, combining online HD map construction with global prior maps significantly boosts performance. For example, HDMapNet [21] improves by 3.0% mAP with SD maps, VectorMapNet [22] gains 3.9% with a neural map prior, and StreamMapNet [63] and MapTRv2 [76] see 5.9% and 5.7% increases with historical rasterized maps. These results highlight the importance of prior maps in enhancing local map inference.

Future research could explore more efficient integration of various prior maps with online HD map construction to optimize performance in complex driving scenarios.

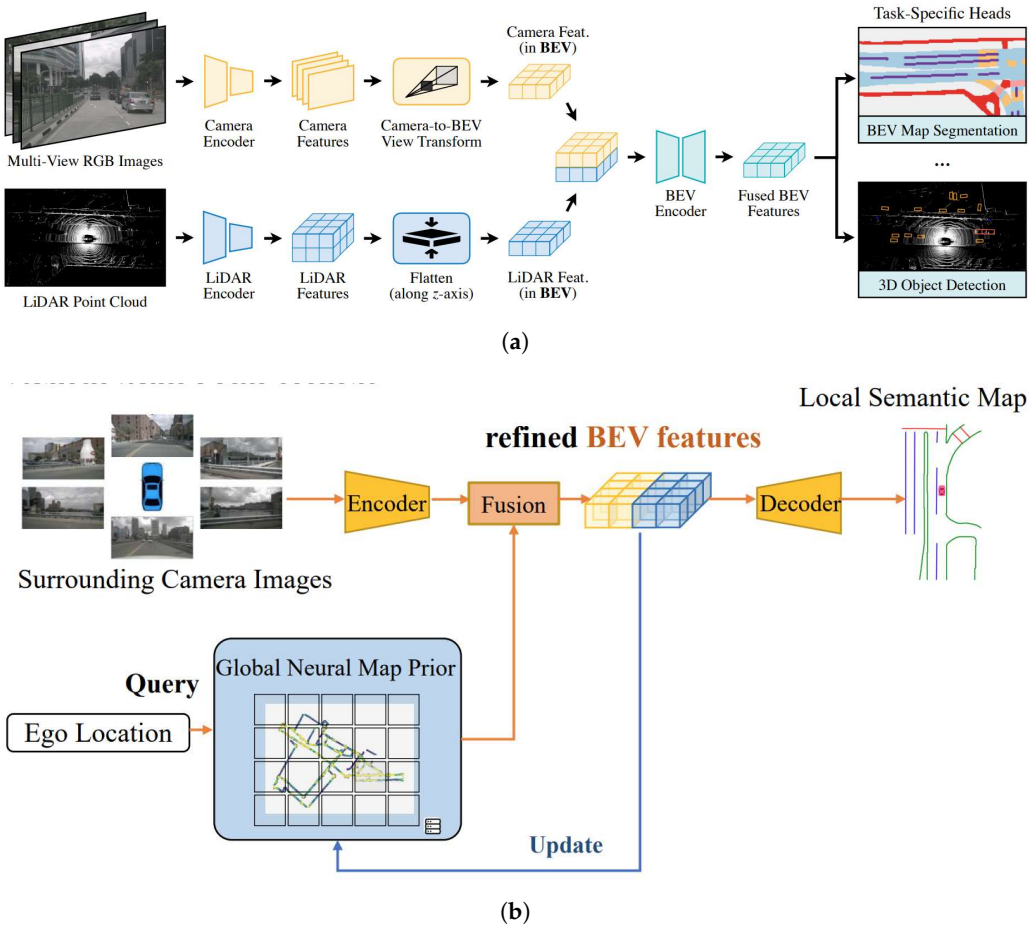


Figure 14. Comparison of the MMF pipeline in two online HD map construction methods. (**a**) BEVFusion [52] fuses camera and LiDAR features in the unified BEV space. (**b**) NMP [58] fuses BEV features with neural map priors from previous traversals.

Table 9. Performance comparison of MMF map element detection methods (prior maps) on the nuScenes [29] validation set. Under **Prior Map**, "SD Map" is standard-definition map [78], "NMP" is neural map prior [58], and "HRMap" is historical rasterized map [70]. The symbol ‡ indicates results from [78], * denotes results from [58], and † represents results from [70].

Method	Prior Map			Map Element Detection			
	SD Map	NMP	HRMap	$AP_{ped.}$	$AP_{div.}$	$AP_{bou.}$	mAP
HDMapNet ‡ [21]	✗	✗	✗	10.3	27.7	45.2	27.7
HDMapNet ‡ [21]	✓	✗	✗	11.3	32.1	48.7	30.7
VectorMapNet * [22]	✗	✗	✗	36.1	47.3	39.3	40.9
VectorMapNet * [22]	✗	✓	✗	42.9	49.6	41.9	44.8
StreamMapNet † [63]	✗	✗	✗	60.4	61.9	58.9	60.4
StreamMapNet † [63]	✗	✗	✓	63.8	69.5	65.5	66.3
MapTRv2 † [76]	✗	✗	✗	59.8	62.4	62.4	61.5
MapTRv2 † [76]	✗	✗	✓	65.8	67.4	68.5	67.2

6. Conclusions

This study provides a comprehensive analysis of online HD map construction, covering task background, high-level motivations, research methodology, key advancements,

existing challenges, and future trends. It is defined as the task of dynamically generating local HD maps from real-time vehicle sensor data, with the potential for improved generalization capability and increased cost efficiency.

The study systematically reviews the latest achievements in online HD map construction, classifying them into three key subtasks: map segmentation, map element detection, and lane graph construction. Specifically, map segmentation methods are classified into projection-based, lift-based, and network-based methods, depending on view transformation techniques from 2D to 3D. Map element detection methods are categorized into CNN-based and Transformer-based methods, based on how the map element point sequences and semantics are decoded. Lane graph construction methods are divided into single-step and iteration-based methods, based on topological reasoning of lane and traffic element relationships. Table 10 presents SWOT analyses of these three sub-tasks, outlining the strengths and weaknesses of each approach.

We conclude that there are still research gaps in online HD map construction, particularly in standardized map representation, multitask learning, and multimodality fusion. The lack of standardized representation can be improved by developing unified representations, incorporating 3D structures and dynamic features, or creating representations that integrate seamlessly with downstream modules. Multitask learning can be advanced by adding perception and prediction tasks or combining multiple map-related tasks. Additionally, multimodality fusion can be enhanced by integrating diverse sensor data or incorporating SD maps and other prior maps.

Table 10. SWOT analyses of three sub-tasks in online HD map construction.

Sub-Task	Strengths	Weaknesses	Opportunities	Threats
Map Segmentation	Generates raster maps with most fine-grained geometric information	Computationally expensive and inefficient; lack of instance-level information of map elements	Coordinates with downstream collision avoidance module; combine multiple view transformation techniques	Possible replacement by 3D semantic scene completion [121,122] or occupancy prediction [123–125]
Map Element Detection	Generates vector maps with instance-level information of map elements; computationally cost-effective	Lower granularity in describing map geometry; lack of topological relationships between instances	Coordinate with downstream collision avoidance module; possible integration with lane graph construction	Possible replacement by end-to-end algorithms [126,127] or visual question answering of traffic scene [128,129]
Lane Graph Construction	Generates vector maps with topological structure of traffic scenes; computationally cost-effective	Needs post-processing to generate clean output; lack of semantic information of map elements	Coordinate with downstream path planning module; combine with knowledge graph of traffic scene [130]	Possible replacement by end-to-end algorithms [126,127] or visual question answering of traffic scene [128,129]

We hope this survey provides valuable insights for researchers and practitioners in the field while also inspiring further exploration of online HD map construction.

Author Contributions: Conceptualization, H.L. and J.S.B.; methodology, H.L., J.S.B., Y.H., K.L., M.S. and S.W.; software, Y.H. and K.L.; validation, H.L. and K.L.; formal analysis, H.L. and K.L.; investigation, H.L. and Y.H.; resources, S.W., M.S. and J.S.B.; data curation, Y.H. and K.L.; writing—original draft preparation, H.L., Y.H. and K.L.; writing—review and editing, S.W., M.S. and J.S.B.; visualization, Y.H. and H.L.; supervision, S.W., M.S. and J.S.B.; project administration, S.W. and

M.S.; funding acquisition, S.W., M.S. and J.S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This study is funded by the Australian Government Research Training Program (RTP) scholarship.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, R.; Wang, J.; Zhang, B. High definition map for automated driving: Overview and analysis. *J. Navig.* **2020**, *73*, 324–341.
2. Fischer, P.; Azimi, S.M.; Roschlaub, R.; Krauß, T. Towards hd maps from aerial imagery: Robust lane marking segmentation using country-scale imagery. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 458.
3. Elghazaly, G.; Frank, R.; Harvey, S.; Safko, S. High-definition maps: Comprehensive survey, challenges and future perspectives. *IEEE Open J. Intell. Transp. Syst.* **2023**, *4*, 527–550.
4. Ebrahimi Soorchaei, B.; Razzaghpour, M.; Valiente, R.; Raftari, A.; Fallah, Y.P. High-definition map representation techniques for automated vehicles. *Electronics* **2022**, *11*, 3374.
5. Diaz-Diaz, A.; Ocaña, M.; Llamazares, Á.; Gómez-Huélamo, C.; Revenga, P.; Bergasa, L.M. Hd maps: Exploiting opendrive potential for path planning and map monitoring. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), IEEE, Aachen, Germany, 5–9 June 2022; pp. 1211–1217.
6. Bučko, B.; Zábovská, K.; Ristvej, J.; Jánošíková, M. HD Maps and Usage of Laser Scanned Data as a Potential Map Layer. In Proceedings of the 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 27 September–1 October 2021; pp. 1670–1675.
7. Bao, Z.; Hossain, S.; Lang, H.; Lin, X. A review of high-definition map creation methods for autonomous driving. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106125.
8. Ilci, V.; Toth, C. High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation. *Sensors* **2020**, *20*, 899.
9. Asrat, K.T.; Cho, H.J. A Comprehensive Survey on High-Definition Map Generation and Maintenance. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 232.
10. Javanmardi, M.; Javanmardi, E.; Gu, Y.; Kamijo, S. Towards high-definition 3D urban mapping: Road feature-based registration of mobile mapping systems and aerial imagery. *Remote Sens.* **2017**, *9*, 975.
11. Massow, K.; Kwella, B.; Pfeifer, N.; Häusler, F.; Pontow, J.; Radusch, I.; Hipp, J.; Dölitzscher, F.; Haueis, M. Deriving HD maps for highly automated driving from vehicular probe data. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1745–1752.
12. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In *Proceedings of the Robotics: Science and Systems*; Carnegie Mellon University Robotics Institute: Berkeley, CA, USA, 2014; Volume 2; pp. 1–9.
13. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, Nevada, USA, 25–29 October 2020; pp. 5135–5142.
14. Yi, S.; Worrall, S.; Nebot, E. Geographical map registration and fusion of lidar-aerial orthoimagery in gis. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 128–134.
15. Huang, X.; Mei, G.; Zhang, J. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11366–11374.
16. Shan, M.; Narula, K.; Worrall, S.; Wong, Y.F.; Perez, J.S.B.; Gray, P.; Nebot, E. A Novel Probabilistic V2X Data Fusion Framework for Cooperative Perception. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 2013–2020.
17. Berrio, J.S.; Shan, M.; Worrall, S.; Nebot, E. Camera-LIDAR integration: Probabilistic sensor fusion for semantic mapping. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 7637–7652.
18. Berrio, J.S.; Worrall, S.; Shan, M.; Nebot, E. Long-term map maintenance pipeline for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10427–10440.
19. Can, Y.B.; Liniger, A.; Paudel, D.P.; Van Gool, L. Structured bird's-eye-view traffic scene understanding from onboard images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 15661–15670.

20. Can, Y.B.; Liniger, A.; Paudel, D.P.; Van Gool, L. Topology preserving local road network estimation from single onboard camera image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Orleans, LA, USA, 18–24 June 2022; pp. 17263–17272.
21. Li, Q.; Wang, Y.; Wang, Y.; Zhao, H. Hdmapnet: An online hd map construction and evaluation framework. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), IEEE, Philadelphia, PA, USA, 19–23 May 2022; pp. 4628–4634.
22. Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y.; Zhao, H. Vectormapnet: End-to-end vectorized hd map learning. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 22352–22369.
23. Tang, X.; Jiang, K.; Yang, M.; Liu, Z.; Jia, P.; Wijaya, B.; Wen, T.; Cui, L.; Yang, D. High-Definition Maps Construction Based on Visual Sensor: A Comprehensive Survey. *IEEE Trans. Intell. Veh.* **2023**, *99*, 1–23, doi:10.1109/TIV.2023.3336940.
24. Ma, Y.; Wang, T.; Bai, X.; Yang, H.; Hou, Y.; Wang, Y.; Qiao, Y.; Yang, R.; Manocha, D.; Zhu, X. Vision-centric bev perception: A survey. *arXiv* **2022**, arXiv:2208.02797.
25. Li, H.; Sima, C.; Dai, J.; Wang, W.; Lu, L.; Wang, H.; Zeng, J.; Li, Z.; Yang, J.; Deng, H.; et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 2151–2170.
26. Murciego, E.; Huélamo, C.G.; Barea, R.; Bergasa, L.M.; Romera, E.; Arango, J.F.; Tradacete, M.; Sáez, Á. Topological road mapping for autonomous driving applications. In Proceedings of the Advances in Physical Agents: Proceedings of the 19th International Workshop of Physical Agents (WAF 2018), Madrid, Spain, 22–23 November 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 257–270.
27. Kwag, J.; Toth, C. A Review on End-to-End High-Definition Map Generation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *48*, 187–194.
28. Ouzzani, M.; Hammady, H.; Fedorowicz, Z.; Elmagarmid, A. Rayyan—A web and mobile app for systematic reviews. *Syst. Rev.* **2016**, *5*, 210.
29. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
30. Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J.K.; et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks, Vistual, 6–14 December 2021.
31. Wang, H.; Li, T.; Li, Y.; Chen, L.; Sima, C.; Liu, Z.; Wang, B.; Jia, P.; Wang, Y.; Jiang, S.; et al. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. In Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks, New Orleans, LA, USA, 10–16 December 2023; IEEE: Piscataway, NJ, USA, 2024; Volume 36.
32. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
33. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8748–8757.
34. Blaschko, M.B.; Lampert, C.H. Learning to Localize Objects with Structured Output Regression. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2008, Marseille, France, 12–18 October 2008*; Forsyth, D., Torr, P., Zisserman, A., Eds.; Springer: Berlin/ Heidelberg, Germany, 2008; pp. 2–15.
35. He, Y.; Bian, C.; Xia, J.; Shi, S.; Yan, Z.; Song, Q.; Xing, G. VI-Map: Infrastructure-Assisted Real-Time HD Mapping for Autonomous Driving. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, ACM MobiCom '23, New York, NY, USA, 2–6 October 2023. https://doi.org/10.1145/3570361.3613280.
36. Peng, R.; Cai, X.; Xu, H.; Lu, J.; Wen, F.; Zhang, W.; Zhang, L. LaneGraph2Seq: Lane Topology Extraction with Language Model via Vertex-Edge Encoding and Connectivity Enhancement. *arXiv* **2024**, arXiv:2401.17609.
37. Liao, Y.; Xie, J.; Geiger, A. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3292–3310.
38. Roddick, T.; Cipolla, R. Predicting semantic map representations from images using pyramid occupancy networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11138–11147.
39. Philion, J.; Fidler, S. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 194–210.
40. Reiher, L.; Lampe, B.; Eckstein, L. 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 Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, Rhodes, Greece, 20–23 September 2020; pp. 1–7.

41. Pan, B.; Sun, J.; Leung, H.Y.T.; Andonian, A.; Zhou, B. Cross-view semantic segmentation for sensing surroundings. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4867–4873.
42. Yang, W.; Li, Q.; Liu, W.; Yu, Y.; Ma, Y.; He, S.; Pan, J. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15536–15545.
43. Saha, A.; Mendez, O.; Russell, C.; Bowden, R. Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021; pp. 5133–5139.
44. Zhou, B.; Krähenbühl, P. Cross-view transformers for real-time map-view semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 13760–13769.
45. Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; Dai, J. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–18.
46. Lu, J.; Zhou, Z.; Zhu, X.; Xu, H.; Zhang, L. Learning ego 3d representation as ray tracing. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2022; pp. 129–144.
47. Bartoccioni, F.; Zablocki, É.; Bursuc, A.; Pérez, P.; Cord, M.; Alahari, K. Lara: Latents and rays for multi-camera bird's-eye-view semantic segmentation. In Proceedings of the Conference on Robot Learning, PMLR, 2023; pp. 1663–1672.
48. Gosala, N.; Valada, A. Bird's-eye-view panoptic segmentation using monocular frontal view images. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1968–1975.
49. Xie, E.; Yu, Z.; Zhou, D.; Philion, J.; Anandkumar, A.; Fidler, S.; Luo, P.; Alvarez, J.M. BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. *arXiv* **2022**, arXiv:2204.05088.
50. Zhang, Y.; Zhu, Z.; Zheng, W.; Huang, J.; Huang, G.; Zhou, J.; Lu, J. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv* **2022**, arXiv:2205.09743.
51. Peng, L.; Chen, Z.; Fu, Z.; Liang, P.; Cheng, E. BEVSegFormer: Bird's Eye View Semantic Segmentation From Arbitrary Camera Rigs. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023; pp. 5935–5943.
52. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.L.; Han, S. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In Proceedings of the 2023 IEEE international conference on robotics and automation (ICRA), IEEE, 2023; pp. 2774–2781.
53. Harley, A.W.; Fang, Z.; Li, J.; Ambrus, R.; Fragkiadaki, K. Simple-bev: What really matters for multi-sensor bev perception? In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023; pp. 2759–2765.
54. Zou, J.; Zhu, Z.; Huang, J.; Yang, T.; Huang, G.; Wang, X. HFT: Lifting Perspective Representations via Hybrid Feature Transformation for BEV Perception. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023; pp. 7046–7053.
55. Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; Zhang, X. Petrv2: A unified framework for 3d perception from multi-camera images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 3262–3272.
56. Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C. MapTR: Structured Modeling and Learning for Online Vectorized HD Map Construction. In Proceedings of the International Conference on Learning Representations, 2023.
57. Qiao, L.; Ding, W.; Qiu, X.; Zhang, C. End-to-End Vectorized HD-Map Construction With Piecewise Bezier Curve. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp. 13218–13228.
58. Xiong, X.; Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y.; Zhao, H. Neural map prior for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp. 17535–17544.
59. Shin, J.; Rameau, F.; Jeong, H.; Kum, D. Instagram: Instance-level graph modeling for vectorized hd map learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2023.
60. Ding, W.; Qiao, L.; Qiu, X.; Zhang, C. Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 3672–3682.
61. Yu, J.; Zhang, Z.; Xia, S.; Sang, J. ScalableMap: Scalable Map Learning for Online Long-Range Vectorized HD Map Construction. In Proceedings of the Conference on Robot Learning, PMLR, 2023; pp. 2429–2443.
62. Zhang, G.; Lin, J.; Wu, S.; Luo, Z.; Xue, Y.; Lu, S.; Wang, Z. Online map vectorization for autonomous driving: A rasterization perspective. *Advances in Neural Information Processing Systems*; 2023; Volume 36.
63. Yuan, T.; Liu, Y.; Wang, Y.; Wang, Y.; Zhao, H. Streammapnet: Streaming mapping network for vectorized online hd map construction. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024; pp. 7356–7365.
64. Liu, R.; Yuan, Z. Compact HD Map Construction via Douglas–Peucker Point Transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, 2024; Volume 38; pp. 3702–3710.
65. Dong, H.; Gu, W.; Zhang, X.; Xu, J.; Ai, R.; Lu, H.; Kannala, J.; Chen, X. Superfusion: Multilevel LiDAR–camera fusion for long-range hd map generation. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024; pp. 9056–9062.

66. Zhou, Y.; Zhang, H.; Yu, J.; Yang, Y.; Jung, S.; Park, S.I.; Yoo, B. HIMap: HybrId Representation Learning for End-to-end Vectorized HD Map Construction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024; pp. 15396–15406.
67. Liu, X.; Wang, S.; Li, W.; Yang, R.; Chen, J.; Zhu, J. Mgmap: Mask-guided learning for online vectorized hd map construction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024; pp. 14812–14821.
68. Xu, Z.; K Wong, K.Y.; Zhao, H. InsMapper: Exploring inner-instance information for vectorized HD mapping. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 296–312.
69. Zhang, Z.; Zhang, Y.; Ding, X.; Jin, F.; Yue, X. Online vectorized hd map construction using geometry. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 73–90.
70. Zhang, X.; Liu, G.; Liu, Z.; Xu, N.; Liu, Y.; Zhao, J. Enhancing vectorized map perception with historical rasterized maps. In Proceedings of the European Conference on Computer Vision, Springer: Berlin/Heidelberg, Germany, 2024; pp. 422–439.
71. Hao, X.; Li, R.; Zhang, H.; Li, D.; Yin, R.; Jung, S.; Park, S.I.; Yoo, B.; Zhao, H.; Zhang, J. Mapdistill: Boosting efficient camera-based hd map construction via camera-lidar fusion model distillation. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 166–183.
72. Wang, S.; Jia, F.; Liu, Y.; Zhao, Y.; Chen, Z.; Wang, T.; Zhang, C.; Zhang, X.; Zhao, F. Stream Query Denoising for Vectorized HD Map Construction. In Proceedings of the European Conference on Computer Vision, 2025.
73. Chen, J.; Wu, Y.; Tan, J.; Ma, H.; Furukawa, Y. Maptracker: Tracking with strided memory fusion for consistent vector hd mapping. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 90–107.
74. Hu, H.; Wang, F.; Wang, Y.; Hu, L.; Xu, J.; Zhang, Z. ADMap: Anti-disturbance framework for reconstructing online vectorized HD map. In Proceedings of the European Conference on Computer Vision, 2025.
75. Liu, Z.; Zhang, X.; Liu, G.; Zhao, J.; Xu, N. Leveraging enhanced queries of point sets for vectorized map construction. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 461–477.
76. Liao, B.; Chen, S.; Zhang, Y.; Jiang, B.; Zhang, Q.; Liu, W.; Huang, C.; Wang, X. Maptrv2: An end-to-end framework for online vectorized hd map construction. In Proceedings of the International Journal of Computer Vision, 2024; pp. 1–23.
77. Li, S.; Lin, J.; Shi, H.; Zhang, J.; Wang, S.; Yao, Y.; Li, Z.; Yang, K. DTCLMapper: Dual Temporal Consistent Learning for Vectorized HD Map Construction. *Trans. Intell. Transport. Sys.* **2024**, *25*, 21672–21686. https://doi.org/10.1109/TITS.2024.3450704.
78. Jiang, Z.; Zhu, Z.; Li, P.; Gao, H.a.; Yuan, T.; Shi, Y.; Zhao, H.; Zhao, H. P-MapNet: Far-Seeing Map Generator Enhanced by Both SDMap and HDMap Priors. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8539–8546. https://doi.org/10.1109/LRA.2024.3447450.
79. Peng, N.; Zhou, X.; Wang, M.; Yang, X.; Chen, S.; Chen, G. PrevPredMap: Exploring Temporal Modeling with Previous Predictions for Online Vectorized HD Map Construction. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024.
80. Xu, Z.; Liu, Y.; Sun, Y.; Liu, M.; Wang, L. Centerlinedet: Centerline graph detection for road lanes with vehicle-mounted sensors by transformer for hd map generation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023; pp. 3553–3559.
81. Can, Y.B.; Liniger, A.; Paudel, D.; Van Gool, L. Online Lane Graph Extraction from Onboard Video. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2023; pp. 1663–1670.
82. Can, Y.B.; Liniger, A.; Paudel, D.; Van Gool, L. Prior Based Online Lane Graph Extraction from Single Onboard Camera Image. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2023; pp. 1671–1678.
83. Can, Y.B.; Liniger, A.; Paudel, D.P.; Van Gool, L. Improving online lane graph extraction by object-lane clustering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 8591–8601.
84. Lu, J.; Peng, R.; Cai, X.; Xu, H.; Li, H.; Wen, F.; Zhang, W.; Zhang, L. Translating Images to Road Network: A Non-Autoregressive Sequence-to-Sequence Approach. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 23–33.
85. Li, T.; Chen, L.; Wang, H.; Li, Y.; Yang, J.; Geng, X.; Jiang, S.; Wang, Y.; Xu, H.; Xu, C.; et al. Graph-based topology reasoning for driving scenes. *arXiv* **2023**, arXiv:2304.05277.
86. Li, T.; Jia, P.; Wang, B.; Chen, L.; Jiang, K.; Yan, J.; Li, H. LaneSegNet: Map Learning with Lane Segment Perception for Autonomous Driving. In Proceedings of the ICLR, 2024.
87. Wu, D.; Chang, J.; Jia, F.; Liu, Y.; Wang, T.; Shen, J. TopoMLP: An Simple yet Strong Pipeline for Driving Topology Reasoning. In Proceedings of the International Conference on Learning Representations, 2024.
88. Luo, K.Z.; Weng, X.; Wang, Y.; Wu, S.; Li, J.; Weinberger, K.Q.; Wang, Y.; Pavone, M. Augmenting lane perception and topology understanding with standard definition navigation maps. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024; pp. 4029–4035.
89. Zhu, T.; Leng, J.; Zhong, J.; Zhang, Z.; Sun, C. Lanemapnet: Lane network recognization and hd map construction using curve region aware temporal bird's-eye-view perception. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2024; pp. 2168–2175.

90. Han, Y.; Yu, K.; Li, Z. Continuity preserving online centerline graph learning. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 342–359.
91. Ma, Z.; Liang, S.; Wen, Y.; Lu, W.; Wan, G. Roadpainter: Points are ideal navigators for topology transformer. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 179–195.
92. Liao, B.; Chen, S.; Jiang, B.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C.; Wang, X. Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 334–351.
93. Kalfaoglu, M.; Ozturk, H.I.; Kilinc, O.; Temizel, A. TopoMaskV2: Enhanced Instance-Mask-Based Formulation for the Road Topology Problem. In Proceedings of the IEEE/CVF European Conference on Computer Vision Workshops, 2024.
94. Fu, Y.; Liao, W.; Liu, X.; Ma, Y.; Dai, F.; Zhang, Y. TopoLogic: An Interpretable Pipeline for Lane Topology Reasoning on Driving Scenes. In *Advances in Neural Information Processing Systems*; 2024.
95. Mao, J.; Shi, S.; Wang, X.; Li, H. 3D object detection for autonomous driving: A comprehensive survey. *Int. J. Comput. Vis.* **2023**, *131*, 1909–1963.
96. Tang, J.; Li, S.; Liu, P. A review of lane detection methods based on deep learning. *Pattern Recognit.* **2021**, *111*, 107623.
97. Mallot, H.A.; Bülthoff, H.H.; Little, J.; Bohrer, S. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cybern.* **1991**, *64*, 177–185.
98. Oliveira, M.; Santos, V.; Sappa, A.D. Multimodal inverse perspective mapping. *Inf. Fusion* **2015**, *24*, 108–121.
99. Bertozz, M.; Broggi, A.; Fascioli, A. Stereo inverse perspective mapping: Theory and applications. *Image Vis. Comput.* **1998**, *16*, 585–590.
100. Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; Wang, C. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4350–4362.
101. Sämann, T.; Amende, K.; Milz, S.; Witt, C.; Simon, M.; Petzold, J. Efficient semantic segmentation for visual bird's-eye view interpretation. In Proceedings of the Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15; Springer: Berlin/Heidelberg, Germany, 2019; pp. 679–688.
102. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
103. Samani, E.U.; Tao, F.; Dasari, H.R.; Ding, S.; Banerjee, A.G. F2BEV: Bird's Eye View Generation from Surround-View Fisheye Camera Images for Automated Driving. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023; pp. 9367–9374.
104. Dutta, P.; Sistu, G.; Yogamani, S.; Galván, E.; McDonald, J. ViT-BEVSeg: A hierarchical transformer network for monocular birds-eye-view segmentation. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022; pp. 1–7.
105. Can, Y.B.; Liniger, A.; Unal, O.; Paudel, D.; Van Gool, L. Understanding bird's-eye view of road semantics using an onboard camera. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3302–3309.
106. Tesla. Tesla AI Day 2021. 2021. Available online: https://www.youtube.com/watch?v=j0z4FweCy4M (accessed on).
107. Lu, C.; van de Molengraft, M.J.G.; Dubbelman, G. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. *IEEE Robot. Autom. Lett.* **2019**, *4*, 445–452.
108. Roddick, T.; Kendall, A.; Cipolla, R. Orthographic feature transform for monocular 3d object detection. *arXiv* **2018**, arXiv:1811.08188.
109. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015; pp. 3431–3440.
110. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
111. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, 2019; pp. 6105–6114.
112. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; pp. 770–778.
113. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; pp. 12697–12705.
114. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337.
115. Liu, Y.; Wang, T.; Zhang, X.; Sun, J. Petr: Position embedding transformation for multi-view 3d object detection. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2022; pp. 531–548.
116. Gu, X.; Song, G.; Gilitschenski, I.; Pavone, M.; Ivanovic, B. Producing and Leveraging Online Map Uncertainty in Trajectory Prediction. *arXiv* **2024**, arXiv:2403.16439.
117. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75.

118. Ranganatha, N.E.; Zhang, H.; Venkatramani, S.; Liao, J.Y.; Christensen, H.I. SemVecNet: Generalizable Vector Map Generation for Arbitrary Sensor Configurations. *arXiv* **2024**, arXiv:2405.00250.
119. Choi, S.; Kim, J.; Shin, H.; Choi, J.W. Mask2map: Vectorized hd map construction using bird's eye view segmentation masks. *arXiv* **2024**, arXiv:2407.13517.
120. Berrio, J.S.; Zhou, W.; Ward, J.; Worrall, S.; Nebot, E. Octree map based on sparse point cloud and heuristic probability distribution for labeled images. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018; pp. 3174–3181. https://doi.org/10.1109/IROS.2018.8594024.
121. Cao, A.Q.; De Charette, R. Monoscene: Monocular 3d semantic scene completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 3991–4001.
122. Li, Y.; Yu, Z.; Choy, C.; Xiao, C.; Alvarez, J.M.; Fidler, S.; Feng, C.; Anandkumar, A. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp. 9087–9098.
123. Zhang, Y.; Zhu, Z.; Du, D. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 9433–9443.
124. Ming, Z.; Berrio, J.S.; Shan, M.; Worrall, S. InverseMatrixVT3D: An Efficient Projection Matrix-Based Approach for 3D Occupancy Prediction. *arXiv* **2024**, arXiv:2401.12422.
125. Ming, Z.; Berrio, J.S.; Shan, M.; Worrall, S. OccFusion: Multi-Sensor Fusion Framework for 3D Semantic Occupancy Prediction. *IEEE Trans. Intell. Veh.* **2024**, *early access*.
126. Casas, S.; Sadat, A.; Urtasun, R. Mp3: A unified model to map, perceive, predict and plan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021; pp. 14403–14412.
127. Chitta, K.; Prakash, A.; Geiger, A. Neat: Neural attention fields for end-to-end autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 15793–15803.
128. Sima, C.; Renz, K.; Chitta, K.; Chen, L.; Zhang, H.; Xie, C.; Beißwenger, J.; Luo, P.; Geiger, A.; Li, H. Drivelm: Driving with graph visual question answering. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2025; pp. 256–274.
129. Marcu, A.M.; Chen, L.; Hünermann, J.; Karnsund, A.; Hanotte, B.; Chidananda, P.; Nair, S.; Badrinarayanan, V.; Kendall, A.; Shotton, J.; et al. LingoQA: Visual question answering for autonomous driving. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2024; pp. 252–269.
130. Guo, Y.; Yin, F.; Li, X.h.; Yan, X.; Xue, T.; Mei, S.; Liu, C.L. Visual traffic knowledge graph generation from scene images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 21604–21613.