Hierarchical Queries for 3D Lane Detection Based on Multi-Frame Point Clouds

Ruixin Liu and Zejian Yuan, *Member, IEEE*

***Abstract*— 3D lane detection based on multi-frame point clouds is a critical task for autonomous driving. The challenge lies in efficiently performing temporal fusion using multiple data frames with incomplete yet complementary contexts. Existing methods either directly concatenate consecutive frames, avoiding intrinsic limitations of the raw data, or fuse entire feature maps, without distinguishing lane-related features from backgrounds. These solutions exhibit room for improvement in both precision and efficiency. In this paper, we propose an end-to-end lane detection network with hierarchical queries, which decodes lane features at different levels in a top-down manner for high-precision localization. This framework can be deployed on multi-frame inputs, as it efficiently achieves lane-related sequence fusion with reduced computational costs and improved inference speed. Specifically, we design semi-parametric lane geometry representations to model lanes as parametric curves and discrete points. Accordingly, hierarchical queries are proposed to focus on two-level lane geometries, including curve queries and point queries. Curve queries capture global structures of lanes projected onto the bird's-eye-view (BEV) flat ground, while point queries aggregate multi-frame sequences obtained through curve-guided sampling, acquiring comprehensive and reliable point-level features. In the training stage, our proposed curve matching and point localization loss optimizes the detected lane geometries at both levels. Experiments conducted on the self-collected MultiBEV dataset validate that our method outperforms previously published single-frame and multi-frame methods. Codes are released at https://github.com/lrx02/HQNet**

***Index Terms*— 3D lane detection, point clouds, multi-frame fusion, hierarchical queries, curve-guided sampling, curve matching, point localization loss.**

I. Introduction

LANE detection, as a critical visual perception task for autonomous driving [1], encompasses both lane localization and instance discrimination. It has gained great attention due to widespread downstream applications, such as lane departure warning [2], lane-keeping assistance [3], high-definition map construction [4], [5], [6], [7], and trajectory predictions [8], [9]. Despite the impressive progress in RGB-based lane detection methods [10], [11], [12], [13], [14], [15], [16], [17], challenges like severe occlusions and information compression due to the camera's inherent optical mechanisms cannot be well resolved, resulting in precision drops and higher false-negative rates. As an alternative, point clouds collected by LiDAR sensors are encoded into the bird's-eye-view (BEV) space, which either enhances 2D lane detection by introducing multi-dimensional information [5], [18], [19] or supports the formulation of high-precision 3D lane detection tasks [20], [21].

However, both single-frame RGB and LiDAR-based approaches still struggle with incomplete contexts under heavy occlusions. Multi-frame data naturally provide spatio-temporal cues, offering a more robust perception by aggregating information across consecutive frames. Current methods focus on efficiently leveraging spatio-temporal cues and complementary visual information to enhance feature representations.

One straightforward idea is to fuse multi-frame information at the data level. Some methods [5], [20], [21] directly concatenate point clouds from consecutive frames and feed them into single-frame lane detectors. These solutions heavily depend on high-quality calibration of vehicle sensors, with large data volumes and computational costs required. Nevertheless, the intrinsic limitations of the raw data have not been fundamentally addressed. Once the input frames are insufficient to fill the contexts, serious localization errors arise, especially in regions with sharp curves and complex topologies.

Other methods [22], [23], [24], [25], [26], [27] propagate inter-frame information at the feature map level. Conv-LSTMs [22] and Conv-GRUs [23] are commonly used to recursively encode time-series features of front-view images, while Transformers [26], [27] provide an attention-based solution by designing queries to integrate temporal information. However, the fusion of excessive features irrelevant to lanes introduces undesirable disturbances and extra computation overhead.

Motivated by these limitations, a natural question arises: *Can we design an efficient feature interaction paradigm that fully exploits available lane information and can be easily extended to accommodate multi-frame inputs?* To this end, we propose HQNet, an end-to-end lane detection network with hierarchical queries. To clarify, multiple frames are categorized as target and auxiliary frames. Auxiliary frames enrich the contexts of target frames, thus enhancing detection results for target frames.

To fully exploit lane-related information, semi-parametric lane geometry representations are designed to simultaneously capture lane structures at both curve and point levels. Parametric curves provide a compact representation that coarsely encodes global lane trends with instance discrimination, while point sets refine the precise localization and local structures. Based on the two-level representations, hierarchical queries are proposed, where curve queries decode curve-level

Received 2 January 2025; revised 15 April 2025; accepted 27 April 2025. Date of publication 9 May 2025; date of current version 16 September 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4704900 and in part by the National Natural Science Foundation of China under Grant 61976170 and Grant 62088102. The Associate Editor for this article was Y. Wiseman. *(Corresponding author: Zejian Yuan.)*

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: sweetylrx@stu.xjtu.edu.cn; yuan.ze.jian@xjtu.edu.cn).

Digital Object Identifier 10.1109/TITS.2025.3565769

information from single-frame inputs, while point queries aggregate lane-related point feature sequences across frames. We design a curve-guided sampling module to associate these two-level features, and the curve matching and point localization loss is adopted to supervise curve parameters and point localization.

Existing public LiDAR point cloud datasets suffer from either insufficient precision [28], [29] or incomplete annotations [30], [31], making them unsuitable for multi-frame lane detection. Given that, we collect the MultiBEV dataset with consecutive frames and high-quality 3D annotations. Comprehensive experiments validate that our method outperforms state-of-the-art works, achieving competitive precision under strict thresholds while maintaining a light model size. The main contributions of this work are threefold:

- We propose semi-parametric lane geometry representations that consistently represent lane curves while flexibly describing point localization.
- A top-down lane detection network with hierarchical queries is presented to achieve efficient sequence-level temporal fusion by exploiting lane-related information.
- A simple yet effective curve matching and point localization loss is designed to supervise curve-level and point-level predictions, balancing coupled shape constraints with flexible localization constraints.

II. Related Work

A. Lane Detection With Different Representations

The majority of pre-existing methods rely on different lane representations, categorized into pixel-level, point-level, and curve-level representations.

Pixel-level representations are used in segmentation-based methods [11], [28], [32], [33], where lanes are represented using semantic pixels. These methods typically produce dense segmentation results without structural modeling, which require post-processing techniques. To reduce computational costs, point-level representations are designed to represent lanes using a series of sparse and flexible points. Point representations in grid-based methods [19], [20], [34], [35], [36] are designed within the divided grid space, combining grid-level segmentation with fine-grained localization. However, it is indispensable to associate lane points with specific instances, which is typically achieved through clustering or other self-designed instance-level discrimination modules. Besides, anchor-based methods [16], [37], [38], [39], [40] work in a top-down manner to regress point offsets with respect to the predefined line anchors. Despite the promising results, their performances highly depend on the predefined anchors, making them dataset-specific. Additionally, predictions far away from anchors may experience a decrease in precision.

In contrast, parametric-based methods [14], [41], [42], [43] directly formulate curve parameter regression problems by representing holistic lane shape as curve-level representations. Such methods achieve smooth and consistent predictions with fast inference speed but have limitations in handling rapid topology changes and precise localization. As improvements, FHLD [21] combines adaptive curve representation with point representation. It detects lane curves with respect to the reference axis and leverages feature blocks around the specific points located on lane curves to refine detection results. Another method [44] adopts a parallel representation, where each lane is represented either as a curve or a set of key points. By comparison, our semi-parametric lane geometry representations utilize a top-down strategy to emphasize geometric consistency against occlusions while preserving local details.

In real-world applications, single-frame data provide only a limited perspective of the scene at any moment, leading to ambiguities in incomplete contexts and topological changes. To ensure a natural extension to multi-frame lane detection tasks, we utilize polynomial parameters to model lane curves and propose HQNet with hierarchical queries to detect lane geometries at both curve and point levels.

B. Multi-Frame Fusion Methods

Multi-frame data provide more comprehensive and reliable visual cues about scenarios. The overlapping and complementary natures of multi-frame input make them widely applicable in object detection [27], [45], [46]. Current methods fuse features at different levels, including data-level fusion, feature-map-level fusion, and more refined fusion methods.

Data-level fusion methods are commonly adopted for LiDAR point cloud data, which concatenate point clouds from consecutive frames to explicitly model inter-frame relationships and enrich the representation space. In this way, single-frame lane detectors such as DAGMapper [5] can be easily extended to multi-frame tasks by recurrently predicting positions and states within the rotated regions of interest. Although this fusion mechanism is simple yet effective, the overwhelming computational costs and high-quality calibration requirements severely limit its feasibility.

As an alternative, feature-map-level fusion methods [22], [23], [24], [25], [26] implicitly construct inter-frame relationships for the entire feature maps through various network architectures, without a heavy dependence on high-quality sensor calibration. Some methods [22], [23] directly employ Conv-LSTMs and Conv-GRUs to integrate spatio-temporal information by stacking features extracted from multiple frames. Besides, MMA-Net [24] designs an LGMA module to attentively aggregate both local and global memory features from other frames. At the same time, TGC-Net [25] introduces a T-RESA module to learn spatio-temporal features along horizontal, vertical, and temporal directions. However, such methods are often computationally expensive and adverse to precision, as they require unnecessary attention on irrelevant regions to lanes. Besides, most of them are designed for 2D lane detection based on front-view images and are not suitable for 3D lane detection based on LiDAR point clouds.

More advanced fusion methods can reduce computation and are typically proposed to address object detection problems. 3D-MAN [45] designs an attention-based multi-view alignment and aggregation module to extract and fuse temporal features according to the box proposals predicted by fast single-frame detectors. DTCLMapper [47]

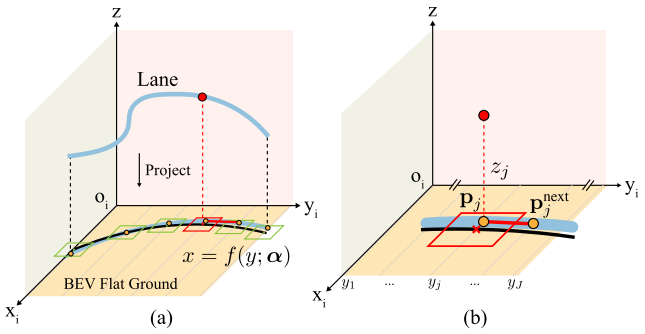

Fig. 1. Semi-parametric lane geometry representations. (a): Parametric curve representation. Each 3D lane (marked in blue) is projected onto the BEV flat ground and depicted by a cubic polynomial function (visualized as the black curve), with the polynomial parameters denoted as $\boldsymbol{\alpha}$. (b): Point with segment shape representation. Each lane point (marked in red) is associated with a 2D position $\mathbf{p}_j$, a height z_j, and a local segment formed by $\mathbf{p}_j$ and $\mathbf{p}_j^{\text{next}}$. Here, x_i-o_i-y_i is the image coordinate system with units in pixels, while the z-axis represents height with units in meters.

and StreamMapNet [48] combine feature-map-level fusion with instance-level fusion, which effectively enhances instance features at a local level while aggregating BEV features at a global level. These dual-level fusion methods facilitate high-definition map construction based on video data with a high overlapping rate. However, there exists a research gap in refined feature fusion tailored for lane detection. HQNet adopts a curve-guided sampling module to sample and reorganize lane-related position and content sequences for each frame. These sequences are then fed into a point decoder to aggregate multi-frame features at a sequence level, avoiding fusion on lane-unrelated backgrounds and preserving crucial lane-related details.

III. Methodology

Inspired by BEV representation learning [49], LiDAR point clouds are projected onto BEV images for lane detection. In Section III-A, we design semi-parametric lane geometry representations to describe lanes at both curve and point levels. Accordingly, an end-to-end lane detection network with hierarchical queries (HQNet) is proposed in Section III-B, which effectively aggregates lane-related sequences and performs high-precision detection based on multi-frame inputs. Section III-C further elaborates on the curve matching and point localization loss to complete our training strategies.

A. Semi-Parametric Lane Geometry Representations

To balance lane structures with high-precision localization, we design novel semi-parametric lane geometry representations that globally model parametric curves in BEV space and locally position non-parametric points in 3D space.

1) Parametric Curve Representation: 3D lanes are projected onto the BEV flat ground for parametric curve modeling, as illustrated in Fig. 1 (a). In detail, the entire projected lane curve (marked in blue) can be approximately modeled by a cubic polynomial function [42] (visualized as the black curve), which has superiority in interpolating the parts without visible cues based on the visible parts. Each curve is defined with respect to the image coordinate system $x_i - o_i - y_i$, written as:

$$x = f(y; \boldsymbol{\alpha}), \tag{1}$$

where $\boldsymbol{\alpha}$ is the set of polynomial parameters.

2) Point With Segment Shape Representation: Due to the limitations of parametric curve representations in capturing subtle changes, we form a point set of size J with equally spaced y-coordinates $Y = \{y_j\}_{j=1}^{J}$ to flexibly describe local lane geometries, which also facilitates the construction of inter-frame correlations. Based on Y, points are sampled from the parametric curves, indicated by red and green 'X' markers, while the boxes represent regions of interest (RoIs), as shown in Fig. 1 (a). The 2D position of the j-th point is denoted as $\mathbf{p}_j = (x_j, y_j)$, and its height as z_j. Additionally, we adopt a segment shape representation similar to [20] and [21] to depict the local structure connected by two lane points, defined as $\mathbf{s}_j = (\mathbf{p}_j, \mathbf{p}_j^{\text{next}})$. As Fig. 1 (b) shows, $\mathbf{p}_j$ and $\mathbf{p}_j^{\text{next}}$ are endpoints of $\mathbf{s}_j$, while the length and heading angle are calculated by $l_j = \|\mathbf{p}_j^{\text{next}} - \mathbf{p}_j\|$ and $\theta_j = \arctan(\frac{y_j^{\text{next}} - y_j}{x_j^{\text{next}} - x_j})$.

To summarize, our proposed semi-parametric lane geometry representations adhere to a top-down strategy that emphasizes the integrity and consistency of lanes, while striving for flexibility and high precision in the local vision.

B. Network Architecture With Hierarchical Queries

The overall network architecture of HQNet is illustrated in Fig. 2. It mainly contains: 1) a shared backbone incorporating convolution neural networks (CNNs) and a Transformer encoder (TRE), 2) a curve decoder (CD) and a point decoder (PD) that utilize hierarchical queries to separately decode feature representations for lane geometry at curve and point levels, and 3) a curve-guided sampling (CGS) module that samples and reorganizes position and content sequences.

1) Lane Detection With Curve Queries: Curve queries are designed for parametric lane curve detection based on single-frame inputs (shown in Fig. 2 left). Given a BEV image $\mathrm{I}^{(t-i)}$ of size $H \times W$ in frame $t-i$, where $i \in \{0, \cdots, T\}$, CNNs and a TRE are employed to capture and integrate contextual information into a global feature map $\mathrm{F}_G^{(t-i)}$. Learnable curve queries $Q_c \in \mathbb{R}^{K \times d}$ are then adopted to interact with $\mathrm{F}_G^{(t-i)}$, where K is the number of predicted lane curves and d is the feature dimension. They are passed through a curve decoder (CD), followed by the curve-level classification and regression heads, denoted as $\mathcal{H}_c$. K sets of polynomial parameters are predicted, with the k-th curve denoted as $\hat{\boldsymbol{\alpha}}_k^{(t-i)}$, accompanied by a confidence score $\hat{c}_k^{(t-i)}$ for the lane or non-lane classification. Here, CD is implemented by a Transformer decoder [50], as adopted in LSTR [42]. The workflow is formulated as:

$$\{(\hat{c}_k^{(t-i)}, \hat{\boldsymbol{\alpha}}_k^{(t-i)})\} = \mathcal{H}_c\left(\mathrm{CD}\left(\mathrm{F}_G^{(t-i)}, Q_c\right)\right). \tag{2}$$

Through parametric curve predictions, global lane structures and instances are consistently captured, serving as a stable guide for sequence sampling.

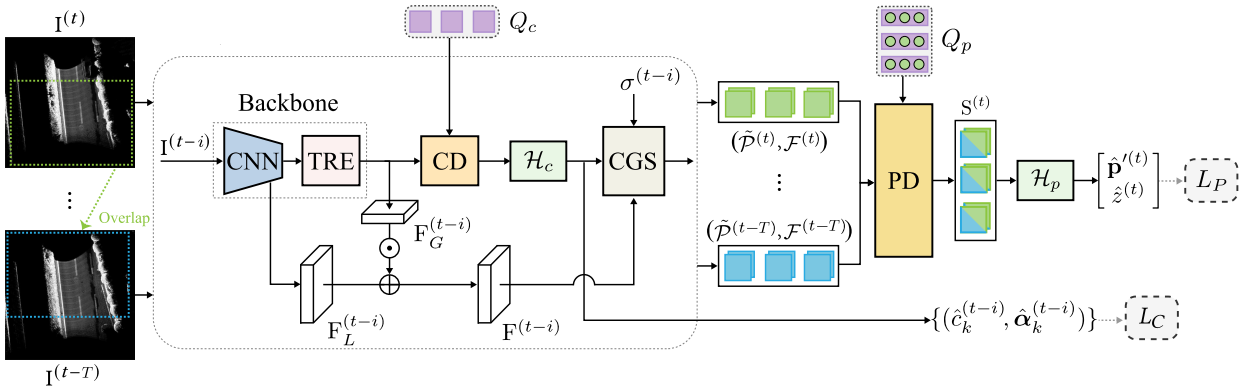

Fig. 2. Network architecture. HQNet starts by extracting features for each frame $I^{(t-i)}$ using a shared backbone that combines convolution neural networks (CNNs) and a Transformer encoder (TRE). Then, curve queries Q_c decode feature representations of lane curves through a curve decoder (CD). Based on the curve parameters regressed by $\mathcal{H}_c$, the curve-guided sampling (CGS) module generates position and content sequences. These procedures are shared across all frames. Finally, point queries Q_p are fed to the point decoder (PD) to perform temporal fusion on multi-frame sequences and predict high-precision point localization with $\mathcal{H}_p$ for the target frame t.

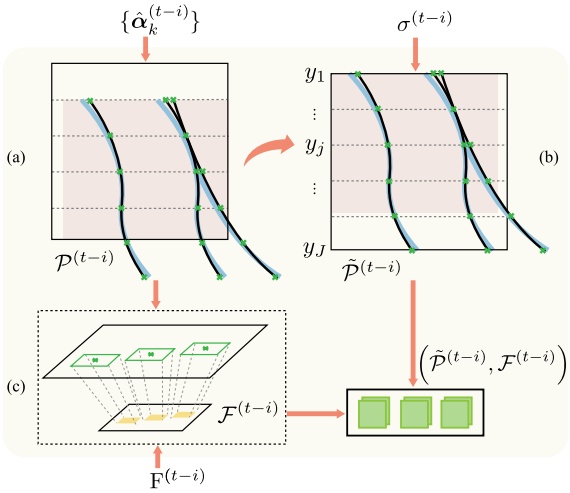

Fig. 3. Curve-guided sampling (CGS). CGS generates position and content sequences for each frame $t-i$ with three operations. (a) Position sampling. Points are sampled from curve predictions, marked as green 'X', generating a position sequence $\mathcal{P}^{(t-i)}$. (b) Position alignment. $\mathcal{P}^{(t-i)}$ is inversely transformed to align with the coordinate system of frame t, generating another position sequence $\tilde{\mathcal{P}}^{(t-i)}$. The overlapping regions are marked by pink. (c) RoI alignment. A content sequence $\mathcal{F}^{(t-i)}$ is extracted through RoIAlign, with RoIs marked as yellow boxes.

2) Curve-Guided Sampling: As demonstrated in Fig. 3, the curve-guided sampling (CGS) module samples and reorganizes position and content sequences under the guidance of curve predictions. This process involves three operations: a) position sampling, b) position alignment, and c) RoI alignment. In our implementation, frame t serves as the target frame, while other frames are set as auxiliary frames. Each frame $t-i$, whether target or auxiliary, obtains a position sequence via position sampling and extracts a content sequence through RoI alignment, based on its respective coordinate system. Subsequently, the position sequence is inversely transformed through position alignment to match the coordinate system of the target frame t.

The workflow of position sampling is shown in Fig. 3 (a). Points are sampled at equal intervals along the y-axis, positioned on the predicted lane curves. Specifically, the shared y-coordinate set, denoted by $Y = \{y_j\}_{j=1}^{J}$, is derived with reference to the coordinate system of the target frame t. Besides, the transformation offset $\sigma^{(t-i)} = (\bar{x}^{(t-i)}, \bar{y}^{(t-i)})$ is calculated based on the vehicle's trajectory from each frame $t-i$ to frame t. Given $y_j + \bar{y}^{(t-i)}$ as the independent variable, the position $\hat{\mathbf{p}}_{j,k}^{(t-i)}$ of the j-th point on the k-th curve in frame $t-i$ can be computed via Equation 1. The sequence of positions is expressed as $\mathcal{P}^{(t-i)} = \{\hat{\mathbf{p}}_{j,k}^{(t-i)}\}$.

To perform positional encoding in a unified coordinate system, a new position sequence $\tilde{\mathcal{P}}^{(t-i)}$ is calculated for position alignment with the target frame t, as shown in Fig. 3 (b). The position sequence $\mathcal{P}^{(t-i)}$ from each frame $t-i$ is inversely transformed to frame t using the transformation offset $\sigma^{(t-i)}$, resulting in the aligned position sequence $\tilde{\mathcal{P}}^{(t-i)}$.

RoI alignment utilizes the position sequence $\mathcal{P}^{(t-i)}$ to obtain the lane-related content sequence, as presented in Fig. 3 (c). Features within fixed-size regions of interest (RoIs) are extracted from the entire feature maps to preserve critical lane information. To ensure localization accuracy, the output of the penultimate layer of the CNN serves as a fine-detailed local feature map $\mathrm{F}_L^{(t-i)}$ (shown in Fig. 2). The global feature map $\mathrm{F}_G^{(t-i)}$ is upsampled to match the shape of $\mathrm{F}_L^{(t-i)}$, and they are added together to form another input to the RoIAlign module, denoted as $\mathrm{F}^{(t-i)} = \bigoplus\left(\bigodot(\mathrm{F}_G^{(t-i)}), \mathrm{F}_L^{(t-i)}\right)$. Here, $\bigodot$ and $\bigoplus$ represent the 'upsample' and 'add' operations, respectively. To focus on RoIs with a fixed size of $r \times r$, the content information $\hat{\mathbf{f}}_{j,k}^{(t-i)}$ is encoded through hard attention from the feature map $\mathrm{F}^{(t-i)}$, following the RoIAlign [51] approach:

$$\hat{\mathbf{f}}_{j,k}^{(t-i)} = \mathrm{RoI}\left(\hat{\mathbf{p}}_{j,k}^{(t-i)}, r, \mathrm{F}^{(t-i)}\right), \tag{3}$$

where $\hat{\mathbf{p}}_{j,k}^{(t-i)}$ is clamped within the image, and the sequence of contents is denoted as $\mathcal{F}^{(t-i)} = \{\hat{\mathbf{f}}_{j,k}^{(t-i)}\}$.

In brief, curve parameter predictions, transformation offsets, and the comprehensive feature map in each frame $t-i$ are fed to CGS, formulated as follows:

$$\left(\tilde{\mathcal{P}}^{(t-i)}, \mathcal{F}^{(t-i)}\right) = \mathrm{CGS}\left(\{\hat{\boldsymbol{\alpha}}_k^{(t-i)}\}, \sigma^{(t-i)}, \mathrm{F}^{(t-i)}\right). \tag{4}$$

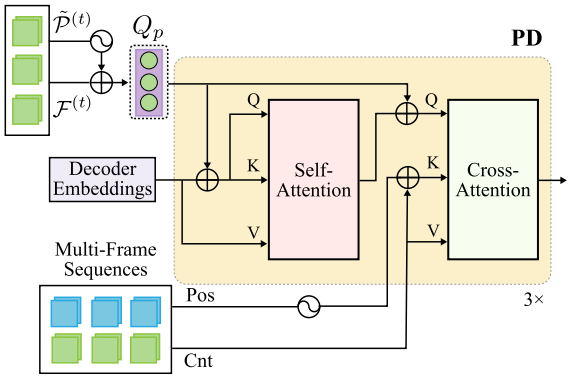

Fig. 4. Point decoder (PD). The encoded position and content sequences in frame t are added together to form point queries Q_p, while the key (K) and value (V) inputs for the cross-attention are derived from the sequences across $T+1$ frames.

By sampling and reorganizing lane-related positions and contents from multiple frames into sequences, excessive backgrounds irrelevant to lanes can be effectively filtered out.

3) High-Precision Localization With Point Queries: Point queries facilitate high-precision lane localization in the target frame t, with a point decoder (PD) enhancing its feature representations. In detail, PD is built upon the vanilla Transformer decoder [50]. Rather than using object queries and spatially encoded feature maps as inputs, point queries derived from frame t and position (Pos) and content (Cnt) sequences from multiple frames are fed to PD to capture inter-frame correlations. As demonstrated in Fig. 4, the Pos and Cnt sequences from $T+1$ frames are concatenated together, and the temporal fusion process is guided by point queries Q_p.

Instead of encoding positions at the feature map level as proposed by [50], we calculate positional encoding only for the points sampled from lane curves, which is uniformly applied to each frame $t-i$ ($i \in \{0, \cdots, T\}$) at the sequence level. The aligned position sequence $\tilde{\mathcal{P}}^{(t-i)}$ is encoded into higher dimensions using sinusoidal functions, denoted as $\mathrm{PE}(\tilde{\mathcal{P}}^{(t-i)})$. This operation standardizes positions across time, providing a consistent reference for multi-frame temporal fusion.

In contrast to learnable queries adopted in previous methods, point queries $Q_p \in \mathbb{R}^{(K\cdot J)\times d}$ are sampled from the curve predictions of the target frame t, capturing both the coarse localization of fitted curves and their surrounding contextual information. Here, Q_p serve as state variables and are dynamically updated during the curve parameter optimization process, which efficiently constrains the output space and stabilizes the final predictions. To obtain Q_p, the position sequence in frame t is encoded and added to its corresponding content sequence, expressed as:

$$Q_p = \mathrm{PE}(\tilde{\mathcal{P}}^{(t)}) + \mathcal{F}^{(t)}. \tag{5}$$

The decoder embeddings (E) are initialized as 0 and share the same size with Q_p, which are projected to the sequence space for information integration.

Besides, the encoded position sequences from $T+1$ frames are concatenated together:

$$\mathrm{Pos} = \mathrm{Cat}(\mathrm{PE}(\tilde{\mathcal{P}}^{(t)}), \cdots, \mathrm{PE}(\tilde{\mathcal{P}}^{(t-T)})), \tag{6}$$

with the corresponding content sequences formed as:

$$\mathrm{Cnt} = \mathrm{Cat}(\mathcal{F}^{(t)}, \cdots, \mathcal{F}^{(t-T)}). \tag{7}$$

Here, "Cat" refers to the concatenation operation. Pos and Cnt are added to serve as keys for the multi-head cross attention, while the values are provided by Cnt. Through sequence-level attention computations, PD reorganizes the fused feature sequences as follows:

$$\mathrm{S}^{(t)} = \mathrm{PD}\left(\{(\tilde{\mathcal{P}}^{(t-i)},\ \mathcal{F}^{(t-i)})\}_{i=0}^{T}, Q_p\right). \tag{8}$$

And the point-level predictions $\left[\hat{\mathbf{p}}'^{(t)}, \hat{z}^{(t)}\right] = \{(\hat{\mathbf{p}}'^{(t)}_{j,k}, \hat{z}^{(t)}_{j,k})\}$ can be generated by a regression head $\mathcal{H}_p$, denoted as:

$$\left[\hat{\mathbf{p}}'^{(t)}, \hat{z}^{(t)}\right] = \mathcal{H}_p\left(\mathrm{S}^{(t)}\right). \tag{9}$$

This sequence-level temporal fusion mechanism employs point queries to aggregate local details across multiple frames, under the guidance of lane-related features.

C. Training With Curve Matching and Point Localization Loss

HQNet is trained based on two-level predictions of lane geometries, with the total loss function formulated as:

$$L_{Total} = L_C + L_P, \tag{10}$$

where the curve matching loss L_C supervises curve parameters, while the point localization loss L_P supervises accurate point positions and their respective segment shapes.

1) Curve Matching Loss: L_C imposes consistent constraints on the overall lane curves by measuring Manhattan distance $D_{\mathrm{Manhattan}}$ between the points densely sampled from ground truths and predictions of parametric representations. Specifically, given the k-th ground truth $\boldsymbol{\alpha}_k^{(t-i)}$ in each frame $t-i$, which matches the $\hat{\epsilon}(k)$-th prediction $\hat{\boldsymbol{\alpha}}_{\hat{\epsilon}(k)}^{(t-i)}$, each point $\mathbf{p}_{j,k}^{(t-i)}$ with its corresponding predicted point $\hat{\mathbf{p}}_{j,\hat{\epsilon}(k)}^{(t-i)}$ is used to calculate the fitting loss L_f, defined as:

$$L_f\left(\hat{\boldsymbol{\alpha}}_{\hat{\epsilon}(k)}^{(t-i)}\right) = \frac{1}{N_k}\sum_{j=1}^{N_k} D_{\mathrm{Manhattan}}\left(\hat{\mathbf{p}}_{j,\hat{\epsilon}(k)}^{(t-i)}, \mathbf{p}_{j,k}^{(t-i)}\right), \tag{11}$$

where N_k denotes the number of densely sampled points, and $\hat{\epsilon}$ is a one-to-one assignment. Besides, the cross-entropy loss is adopted to supervise the existence confidences of lane curves, denoted as L_{CE}. The curve matching loss is formulated as:

$$\begin{aligned} L_C = \frac{1}{T+1}\sum_{i=0}^{T}\sum_{k=1}^{K}\big(&\lambda_1 L_{CE}(\hat{c}_{\hat{\epsilon}(k)}^{(t-i)}, c_k^{(t-i)}) \\ &+ \mathbb{1}(c_k^{(t-i)} \neq 0)\lambda_2 L_f\left(\hat{\boldsymbol{\alpha}}_{\hat{\epsilon}(k)}^{(t-i)}\right)\big), \end{aligned} \tag{12}$$

where $\mathbb{1}(\cdot)$ is an indicator function, $c_k^{(t-i)} \in \{0, 1\}$ represents the label of the k-th curve, and λ_1 and λ_2 are loss coefficients.

The aforementioned assignment $\hat{\epsilon}$ is obtained by formulating a bipartite matching problem, as presented in LSTR [42], with the matching cost defined by classification and localization terms. Hungarian algorithm [52] is then applied to determine the optimal assignment $\hat{\epsilon}$ by minimizing the overall cost.

2) Point Localization Loss: L_P is only enforced on the target frame t, defined by two parts:

$$L_P = \lambda_3 L_{loc} + \lambda_4 L_{seg}, \tag{13}$$

where the localization loss L_{loc} and the segment shape loss L_{seg} supervise the localization of isolated points and shapes of coupled segments, respectively. λ_3 and λ_4 are loss coefficients.

Since curve predictions are dynamically updated, L_{loc} calculates the smooth L1 loss to supervise the absolute positions and heights of the sampled points. Only the matched lane curves are adopted for point-level supervision, denoted as:

$$L_{loc} = \sum_{k=1}^{K} \mathbb{1}\left(c_k^{(t)} \neq 0\right) \sum_{j=1}^{J} \times \left(L_{smooth\text{-}l_1}\left(\hat{\mathbf{p}}'^{(t)}_{j,\hat{\epsilon}(k)}, \mathbf{p}^{(t)}_{j,k}\right) + L_{smooth\text{-}l_1}\left(\hat{z}^{(t)}_{j,\hat{\epsilon}(k)}, z^{(t)}_{j,k}\right)\right). \tag{14}$$

To improve localization precision, L_{seg} jointly optimizes the coupled shape parameters of each segment $\mathbf{s}$ as proposed by [53]. Specifically, we convert $\mathbf{s}$ to a 2D Gaussian distribution $\mathcal{N}$ following [20], [21]. For the j-th segment on the $\hat{\epsilon}(k)$-th curve in frame t, the predicted Gaussian distribution $\hat{\mathcal{N}}^{(t)}_{j,\hat{\epsilon}(k)}$ and its ground truth $\mathcal{N}^{(t)}_{j,k}$ are forced together using the symmetric Kullback-Leibler divergence (KLD), written as:

$$L_{seg} = \frac{1}{2} \sum_{k=1}^{K} \mathbb{1}\left(c_k^{(t)} \neq 0\right) \sum_{j=1}^{J-1} \times \left(\mathrm{KLD}\left(\hat{\mathcal{N}}^{(t)}_{j,\hat{\epsilon}(k)}, \mathcal{N}^{(t)}_{j,k}\right) \mathrm{KLD}\left(\mathcal{N}^{(t)}_{j,k}, \hat{\mathcal{N}}^{(t)}_{j,\hat{\epsilon}(k)}\right)\right). \tag{15}$$

3) Training Strategies: During training, a two-stage strategy is adopted to stabilize the optimization process. In the early stage, to prevent interference between hierarchical lane geometries, L_P is not optimized for the first 10 epochs. Subsequently, the entire network is jointly trained with both L_C and L_P to seek a better precision performance.

IV. Experiments

A. Dataset

Recently, there are no publicly accessible multi-frame point cloud benchmark datasets dedicated to lane detection tasks. Pre-existing datasets are either unqualified for high-definition map construction [28] or inadequately annotated [30], [31]. Therefore, we conduct experiments on a self-collected MultiBEV dataset, which comprises consecutive, highly precise, and dense point cloud frames, and provides high-quality 3D annotations. To be specific, we use vehicles equipped with high-precision LiDAR, GPS, and IMU sensors to gather both raw data and vehicle trajectory information.

The MultiBEV dataset collects approximately 200 km of road data from densely populated cities, including roughly 20% challenging scenes (e.g., severe road marking degradation, heavy holes caused by occlusions, diverse topology changes, and margins with sparse points) and different lane types (55% solid lanes, 35% dotted lanes, and other lane types). Each data frame covers a drivable area of approximately 25 × 25 m, centered on the vehicle's current position for cropping. The frame is rasterized and vertically projected into a BEV image with a resolution of 800 × 800.

We annotate each lane instance using a sequence of 3D points located on its centerline from the start to the end, along with the corresponding instance index and lane type. Annotations are made at intervals of around 0.5 meters, with extra points annotated on curved lanes. In a single frame, the number of lanes ranges from 1 to 9. MultiBEV contains a total of around 25k images, which are split into 80% for training and 20% for testing. More details are provided in the supplementary materials.

B. Evaluation Metrics

We define the metrics precision (%), recall (%), and F1 score (%) in a rigorous centimeter-level manner, consistent with the approach used in DSANet [20] and FHLD [21]. To accomplish this, both predicted and ground-truth lane points are densely interpolated to search the number of points that fall within the specific distance thresholds. Our experiments focus on physical distance thresholds of 5 cm, 10 cm, and 20 cm, respectively. Efficiency metrics are reported, with the FPS calculated based on a batch size of 1 using the Nvidia RTX 3090 implemented in Pytorch. Furthermore, the total number of model parameters (Params) and the multiply-accumulate operations (MACs) are presented.

C. Implementation Details

BEV images are normalized and resized to $(H, W) = (640, 640)$, serving as inputs to HQNet. We adopt the commonly used ResNet18 [54], with the output channels of each block reduced to "16, 32, 64, 128" to avoid overfitting. The number of input frames is set to 3. The numbers of predicted lane curves and points are set to $K = 15$ and $J = 40$, respectively. Additionally, the size of RoIs is set to $r = 36$. The models are trained for 200 epochs on a single Nvidia RTX 3090 with a batch size of 8 per frame, using the AdamW optimizer [55] with an initial learning rate of 0.0002 and cosine annealing schedules. The loss coefficients λ_1, λ_2, λ_3, and λ_4 are set to 1, 1.6667, 1, and 0.1, respectively.

D. Comparisons With State-of-the-Art Methods

1) Baselines: DSANet [20] and FHLD [21] are two kinds of advanced solutions for lane detection based on single-frame LiDAR point cloud data. Due to the scarcity of baselines with available codes and datasets, we adapt several front-view-based solutions as additional competitors, including both single-frame methods (CondLaneNet [35], Lane-ATT [39], PolyLaneNet [41], and LSTR [42]) and multi-frame methods (SegNet-ConvLSTM [22] and UNet-ConvLSTM [22]).

For single-frame methods, LSTR[1] requires adaptation for polynomial curve modeling when migrating from the front view to BEV, and Lane-ATT requires pre-designed anchors in experiments. For multi-frame methods, segmentation-based

[1]The adapted polynomial curve modeling of LSTR is equivalent to that of HQNet. More details of baseline setups can be found in the supplementary materials.

TABLE I

COMPARISONS OF PRECISION (%), RECALL (%), AND F1 SCORE (%) ON THE MULTIBEV TESTING SET. THE TOTAL NUMBER OF MODEL PARAMETERS (PARAMS) IS REPORTED IN MILLIONS (M)

Input	Method	Results (5 cm)			Results (10 cm)			Results (20 cm)			MACs	Params	FPS
		P	R	F1	P	R	F1	P	R	F1			
Single-Frame	CondLaneNet [35]	61.58	62.90	62.23	83.16	84.73	83.94	92.15	93.65	92.89	16.29 G	11.93	98
	Lane-ATT [39]	60.15	58.57	59.35	78.38	76.29	77.32	87.25	84.91	86.06	16.33 G	12.68	136
	PolyLaneNet [41]	35.78	35.48	35.63	56.91	56.49	56.70	74.35	74.07	74.21	29.97 G	21.32	127
	LSTR [42]	52.99	52.12	52.55	76.96	75.74	76.35	89.81	88.48	89.14	1.00 G	0.76	143
	DSANet [20]	62.26	61.12	61.68	81.29	79.90	80.59	88.95	87.91	88.43	4.27 G	6.95	12
	FHLD [21]	60.21	56.58	58.34	85.01	80.10	82.48	93.59	88.67	91.06	2.05 G	3.20	93
Multi-Frame	SegNet-ConvLSTM [22]	51.50	54.37	52.90	69.46	73.37	71.36	79.96	84.61	82.22	1.68 T	67.20	14
	UNet-ConvLSTM [22]	56.84	57.14	56.99	74.73	75.18	74.95	83.77	84.41	84.09	558.77 G	51.15	17
	HQNet (Ours)	**75.03**	**75.40**	**75.21**	**88.64**	**88.92**	**88.78**	**93.92**	**93.88**	**93.90**	3.75 G	2.32	33

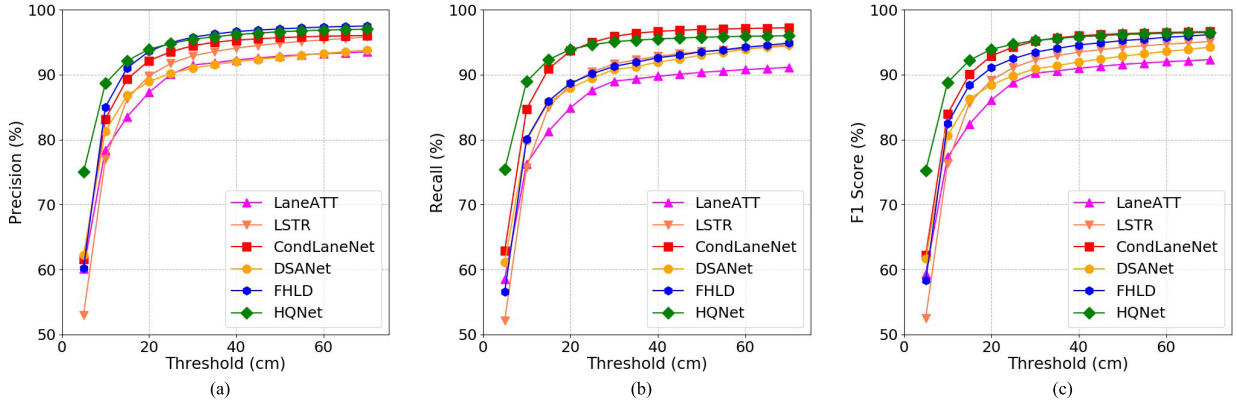

Fig. 5. Performance comparisons of HQNet and its competitors under more thresholds. (a): Precision. (b): Recall. (c): F1 score.

designs are not well suited to our MultiBEV dataset, thereby requiring additional configurations to align with our evaluation metrics. Specifically, SegNet-ConvLSTM and UNet-ConvLSTM need to convert annotations into binary segmentation maps. Besides, post-processing techniques are involved during evaluation to achieve instance discrimination and derive point predictions. Given the challenges some comparative methods face in extending into 3D lane detection, we directly set height predictions as ground truths for fair comparisons.

2) Quantitative Comparisons: In Table I, we present quantitative results for all comparative methods applied to the MultiBEV testing dataset, including precision, recall, and F1 scores under various thresholds. Compared to single-frame methods, HQNet outperforms the second best-performing CondLaneNet, achieving F1 score improvements of 12.98%, 4.84%, and 1.01% under thresholds of 5 cm, 10 cm, and 20 cm. Additionally, HQNet exceeds the competitive DSANet by 12.77% in precision at 5 cm, and surpasses the next best FHLD by 3.63% and 0.33% in precision at 10 cm and 20 cm, respectively. Fig. 5 demonstrates more performance comparisons under different thresholds. DSANet's performance is constrained by the accuracy of the post-clustering process, while FHLD's adaptive reference axes for lane modeling and the point localization based on the local scope are unsuitable for handling incomplete data affected by view truncation.

Compared to multi-frame methods, HQNet outperforms the segmentation-based UNet-ConvLSTM, achieving notable increases of 18.22%, 13.83%, and 9.81% in F1 scores, along with a 2× improvement in inference speed. Considering the limitations of segmentation-based methods in point-level evaluation, we further validate the effectiveness of the fusion mechanism in the subsequent ablation studies.

3) Qualitative Comparisons: Fig. 6 demonstrates visualization results of comparative lane detectors, including (a) dense lines, (b) severe holes, (c) forks, and (d) curves. The rightmost column indicates that HQNet delivers complete and accurate lane estimation results, with correct lane topologies and high-precision lane point locations. Owing to parametric curve predictions, our HQNet robustly performs curve fitting even in regions where lane markings are invisible, such as areas distant from the vehicle sensor and holes caused by occlusions. Meanwhile, point queries efficiently interact with inter-lane and intra-lane sequences across different frames, refining lane point localization.

In comparison, CondLaneNet provides flexible predictions but performs poorly when lanes are located on the road surface with broken content information (CondLaneNet (b), (d)). Moreover, it may experience a performance drop when encountering dense lines or forks (CondLaneNet (a), (c)). Lane-ATT heavily relies on preset anchors, resulting in false negatives (Lane-ATT (a), (c)). Additionally, the straight-line anchors are not adaptive to curves, leading to reduced accuracy as the distance from the anchors increases (Lane-ATT (d)). As for LSTR, it models lane curves using polynomial parameters, making it robust even when partial contexts are lacking (LSTR (b)). Nevertheless, its sensitivity to parameter

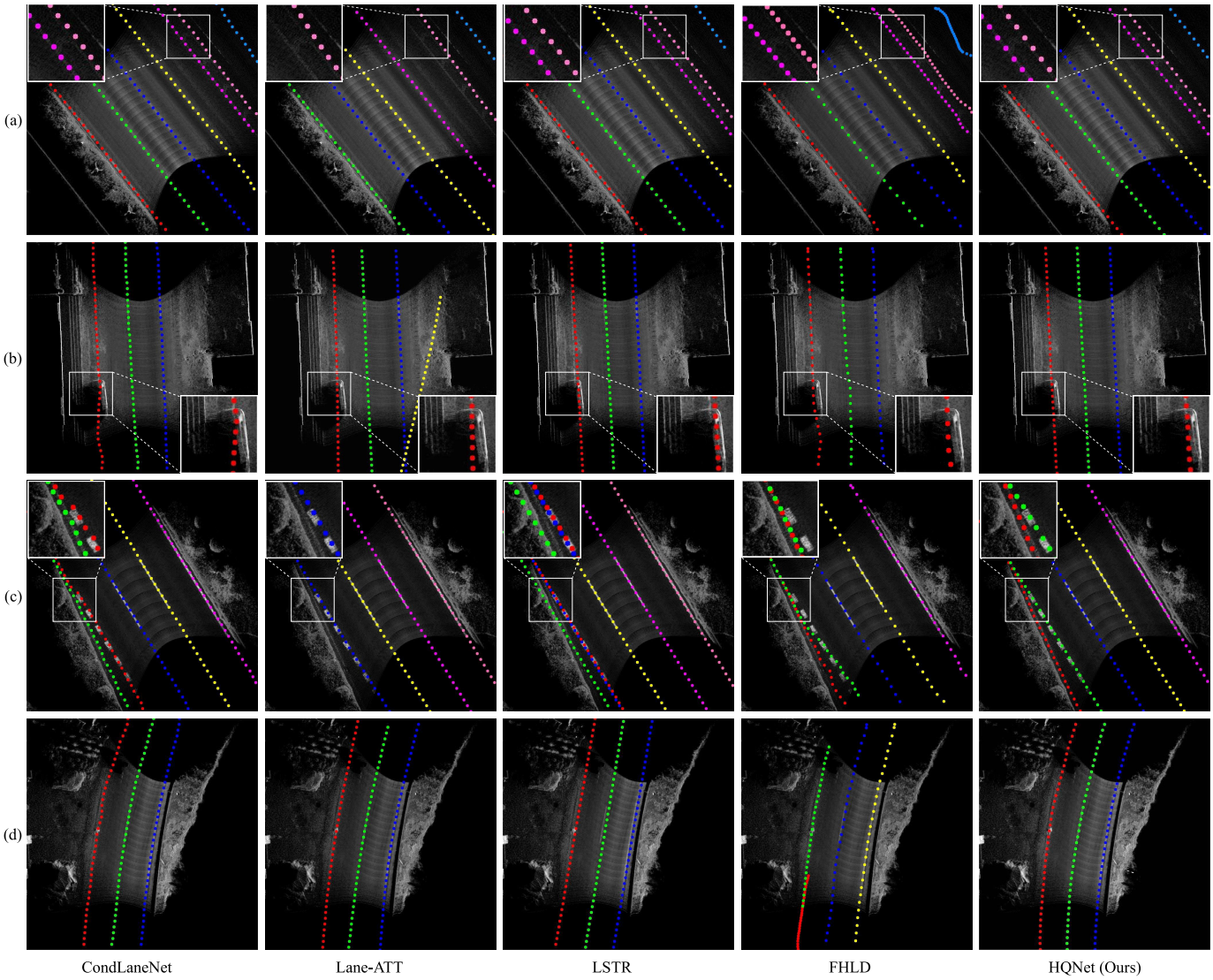

Fig. 6. Visualization results of comparative lane detectors. (a): Dense lines. (b): Severe holes. (c): Forks. (d): Curves. Best viewed in colors.

optimization sacrifices both flexibility and precision in local scopes (LSTR (b), (c), (d)). FHLD achieves high-precision detection in most regions with rich contents (FHLD (a), (b)), however, the independent point offset regressions within local grids may perform worse than its parametric predictions, especially when handling incomplete contents. And the incorrect starting/ending point detection may lead to errors in lane instance discrimination (FHLD (d)).

4) Failure Cases: Subject to the limitations of the top-down framework, HQNet may lead to error detection, as its performance heavily relies on the quality of curve predictions. Typical failure cases are given in Fig. 7. In Fig. 7 (a), we observe false negatives arising from inaccurate curve matching results, which remain uncorrected in subsequent predictions. A possible improvement is to delay the matching process until after point-level predictions, incorporating the refined point localization into the matching process to correct curve matching errors. Fig. 7 (b) shows a precision drop caused by lane marking degradation. It cannot be improved by point queries due to the ambiguity of local features along the whole lane. A potential improvement is to develop a data augmentation strategy that simulates physical degradation, taking rare scenarios [56] into account. Besides, integrating camera images for multi-sensor fusion could further improve robustness.

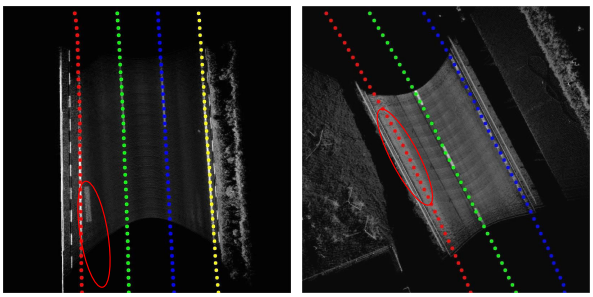

Fig. 7. Failure cases. (a): A false negative arising from inaccurate curve matching. (b): A precision drop caused by lane marking degradation.

TABLE II
ABLATION STUDIES ON COMPONENTS OF OUR NETWORK

Frame	Query			Fusion		Results (5 cm)			Results (10 cm)			Results (20 cm)		
	Q_c	Q_p		Map	Seq	P	R	F1	P	R	F1	P	R	F1
		pos	cnt											
1	✓					61.05	61.17	61.11	79.97	79.94	79.95	89.08	88.72	88.90
	✓	✓	✓			73.74	75.73	74.72	86.21	88.43	87.31	91.11	93.27	92.18
3	✓			✓		68.49	69.39	68.94	85.07	86.00	85.53	91.88	92.51	92.19
	✓	✓			✓	71.63	74.84	73.20	84.24	87.92	86.04	89.20	92.90	91.01
	✓	✓	✓		✓	**75.03**	**75.40**	**75.21**	**88.64**	**88.92**	**88.78**	**93.92**	**93.88**	**93.90**

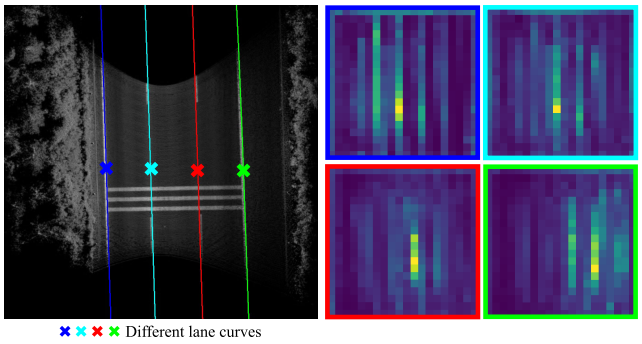

Fig. 8. Attention weight visualization for curve queries. Lane curves are depicted by lines with 'X' symbols in various colors. The corresponding attention weight maps use the same color schemes. Brighter areas represent higher attention weights.

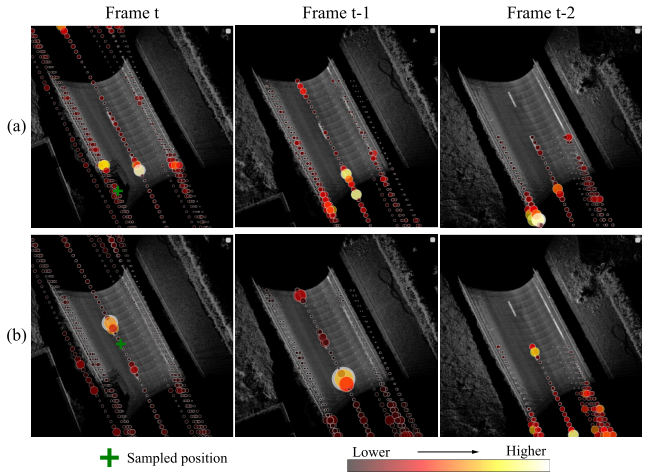

Fig. 9. Attention weight visualization for point queries. Green '+' symbols indicate the positions sampled for point queries, which are used to calculate attention weights across multi-frame sequences. Brighter circles with larger radii represent higher attention weights, while darker circles with smaller radii represent lower attention weights.

E. Ablation Studies

We carry out extensive ablation studies to validate the effectiveness of hierarchical queries, the impact of different feature fusion mechanisms, the influence of varying numbers of input frames and sampled points, as well as the importance of the segment shape loss. Additional ablation studies and qualitative results are included in the supplementary materials.

1) Effectiveness of Hierarchical Queries: Comparisons are conducted between using only curve queries and using both curve and point queries. We formulate a network variant that first regresses lane curve parameters using curve queries, followed by convolution layers in place of point queries to achieve point localization. Due to the impracticality of using only curve queries for multi-frame fusion, comparative results are reported based on single-frame inputs, as detailed in Table II. HQNet employs Q_c and Q_p to perform lane detection based on single-frame inputs, which outperforms the use of Q_c by 13.61%/7.36%/3.28% in F1 scores under thresholds of 5 cm, 10 cm, and 20 cm, respectively. To further evaluate the role of the content (cnt) sequence in point queries, we retain only the position (pos) sequence for comparison. Removing the content sequence leads to a significant drop in precision (shown in Table II, rows 4 and 5), as both the position and content sequences provide essential prior information.

The attention weights of curve queries are visualized in Fig. 8. We observe that the attention mainly concentrates on the structure of lane curves, enabling the distinction of lane-related information from the entire feature map. Besides, the visualization of attention weights for point queries is demonstrated in Fig. 9, where brighter circles with larger radii indicate higher attention and darker circles with smaller radii indicate lower attention. As shown in Fig. 9 (a), when handling regions with severe holes, attention tends to focus on neighboring lane curves in frame t and complementary information provided by other frames. For positions without lane markings, attention is mainly directed towards their adjacent lane markings, as Fig. 9 (b) illustrates. In both scenarios, lower attention is assigned to negative curves that deviate from lanes, validating the effectiveness of our sequence-level fusion.

2) Impact of Different Feature Fusion Mechanisms: To further elaborate on the superiority of our sequence-level fusion mechanism, we present the evaluation results of adopting the feature-map-level fusion (**Map**) and the sequence-level fusion (**Seq**) in Table II. Specifically, the feature-map-level fusion is implemented in a Transformer decoder architecture. The network first extracts a feature map for each frame and then integrates information from multi-frame feature maps to generate a fused feature map. Based on the fused feature map, curve queries are employed to predict lane curve parameters, and lane point locations are regressed through convolution layers accordingly. The sequence-level fusion mechanism used in our proposed HQNet shows improvements of 6.27%/3.25%/1.71% in F1 scores over the feature-map-level fusion, which is attributed to the integration of lane-related features that effectively excludes most disturbances.

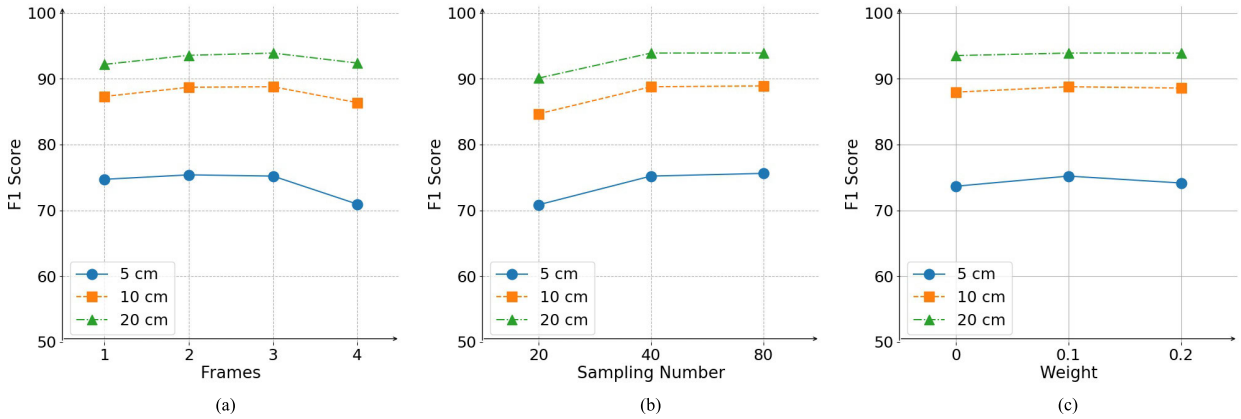

Fig. 10. Ablation studies on components of HQNet. (a): Varying numbers of input frames. (b): Different numbers of sampled points. (c): Effect of the segment shape loss.

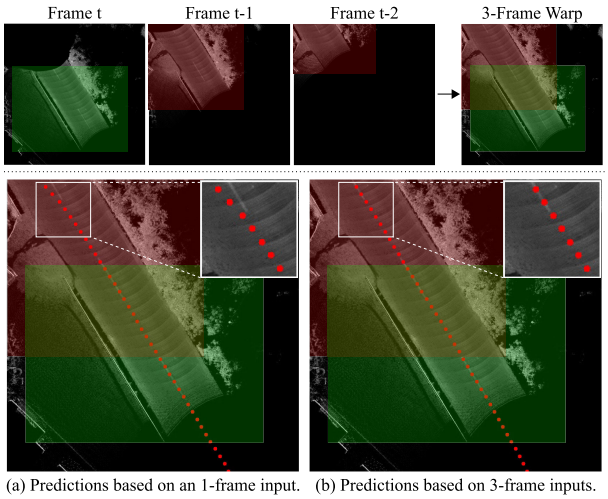

Fig. 11. Visual comparisons of varying numbers of input frames. The upper row shows the valid complementary contexts provided by target frame t (marked in green) and the auxiliary frames $t-1$ and $t-2$ (marked in red). The lower row presents predictions based on different numbers of frames.

3) Varying Numbers of Input Frames: We compare the influence of different frames on sequence-level temporal fusion. Since the MultiBEV dataset has already reached saturation with up to 3-frame inputs, we conduct ablation studies with frame numbers ranging from 1 to 4. Results are detailed in Fig. 10 (a). We observe that the F1 score initially improves as the number of input frames increases, peaking at 3 frames. However, when the frame number reaches 4, performance decreases due to the use of non-complementary contexts for curve predictions, which introduces inconsistencies in the target frame.

Fig. 11 provides straightforward visual comparisons of varying numbers of input frames. The upper row demonstrates a unified virtual view that warps three consecutive frames. The first column presents a visualization of frame t, which serves as the coordinate system reference. The second and third columns show the complementary information supplemented by frames $t-1$ and $t-2$, which are aligned with frame t and constrained within its representation range. The last column visualizes the concatenation of three frames based on point clouds. Additionally, the lower row presents a comparison between predictions obtained from an 1-frame input and those from 3-frame inputs. We notice that HQNet performs well in regions with complete contexts, regardless of whether 1-frame or 3-frame inputs are adopted. However, in the presence of insufficient information, the 1-frame input exhibits a precision drop (Fig. 11 (a)), while predictions based on 3-frame inputs remain stable and precise (Fig. 11 (b)).

4) Different Numbers of Sampled Points: We also conduct experiments on different numbers of sampled points, denoted as J. Results are illustrated in Fig. 10 (b). When J is set to 20, the sparser sampled points may skip key lane-related features, resulting in a decrease in performance, especially near the endpoints. When J gradually increases to 80, the denser sampled points produce more refined localization. However, the smaller lane segments formed by adjacent points dilute the effect of local shape constraints. Considering computational costs, J is set as 40.

5) Importance of the Segment Shape Loss: Experiments are carried out to validate the effect of L_{seg}, analyzing how the weight λ_4 influences the model's performances. Results are demonstrated in Fig. 10 (c). Compared to directly imposing absolute position constraints (i.e. $\lambda_4 = 0$), setting an appropriate weight of $\lambda_4 = 0.1$ achieves an improvement in F1 scores. However, an excessive weight of $\lambda_4 = 0.2$ shifts the optimization focus away from position refinement, negatively impacting the performance.

V. Conclusion

In this work, we propose HQNet, an end-to-end lane detection network that employs hierarchical queries to focus on two levels of lane geometries. The hierarchical queries efficiently aggregate features in a top-down manner, preserving lane curve instances while improving lane localization. This framework performs well in both single-frame and multi-frame lane detection. HQNet is thoroughly validated on the MultiBEV dataset, demonstrating a precision improvement over state-of-the-art methods, achieving the lightest model size and the fastest FPS compared to other multi-frame methods. Our method can be easily extended to other applications, such as road edge detection and high-definition map construction.

Moreover, integrating multi-modal data (e.g., point clouds and camera images) provides a promising direction for enhancing the robustness and precision of long-range lane detection, and will be explored in future work.

REFERENCES

[1] J. M. Jordan, "Autonomous vehicles," in *Robots*. Hershey, PA, USA: IGI Global, 2016, ch. 1, pp. 97–132.

[2] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.

[3] R. Song and B. Li, "Surrounding vehicles' lane change maneuver prediction and detection for intelligent vehicles: A comprehensive review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6046–6062, Jul. 2022.

[4] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "HD maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3611–3619.

[5] N. Homayounfar, J. Liang, W.-C. Ma, J. Fan, X. Wu, and R. Urtasun, "DAGMapper: Learning to map by discovering lane topology," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2911–2920.

[6] R. Liu and Z. Yuan, "Compact HD map construction via douglas-peucker point transformer," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 4, pp. 3702–3710.

[7] B. Liao et al., "MapTRv2: An end-to-end framework for online vectorized HD map construction," *Int. J. Comput. Vis.*, vol. 133, no. 3, pp. 1352–1374, Mar. 2025.

[8] X. Chen, J. Zheng, C. Li, B. Wu, H. Wu, and J. Montewka, "Maritime traffic situation awareness analysis via high-fidelity ship imaging trajectory," *Multimedia Tools Appl.*, vol. 83, no. 16, pp. 48907–48923, Nov. 2023.

[9] S. Zhou, I. Lashkov, H. Xu, G. Zhang, and Y. Yang, "Optimized long short-term memory network for LiDAR-based vehicle trajectory prediction through Bayesian optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 3, pp. 2988–3003, Mar. 2025.

[10] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3261–3264.

[11] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–21.

[12] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[13] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on local: Detecting lane marker from bottom up via key point," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14122–14130.

[14] Z. Feng, S. Guo, X. Tan, K. Xu, M. Wang, and L. Ma, "Rethinking efficient lane detection via curve modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17062–17070.

[15] H. Honda and Y. Uchida, "CLRerNet: Improving confidence of lane detection with LaneIoU," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2024, pp. 1176–1185.

[16] S. Huang et al., "Anchor3DLane++: 3D lane detection via sample-adaptive sparse 3D anchor regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 3, pp. 1660–1673, Mar. 2025.

[17] J. Zhao, Z. Qiu, H. Hu, and S. Sun, "HWLane: HW-transformer for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 9321–9331, Aug. 2024.

[18] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun, "Deep multi-sensor lane detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3102–3109.

[19] N. Homayounfar, W.-C. Ma, S. K. Lakshmikanth, and R. Urtasun, "Hierarchical recurrent attention networks for structured online maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3417–3426.

[20] R. Liu et al., "Learning to detect 3D lanes by shape matching and embedding," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4280–4288.

[21] Z. Guan et al., "Flexible 3D lane detection by hierarchical shape matching," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 1, pp. 694–701.

[22] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020.

[23] J. Zhang, T. Deng, F. Yan, and W. Liu, "Lane detection model based on spatio-temporal network with double convolutional gated recurrent units," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6666–6678, Jul. 2022.

[24] Y. Zhang et al., "VIL-100: A new dataset and a baseline model for video instance lane detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15661–15670.

[25] M. Wang, Y. Zhang, W. Feng, L. Zhu, and S. Wang, "Video instance lane detection via deep temporal and geometry consistency constraints," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2324–2332.

[26] H. Zhang, Y. Gu, X. Wang, J. Pan, and M. Wang, "Lane detection transformer based on multi-frame horizontal and vertical attention and visual transformer module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2022, pp. 1–16.

[27] Z. Luo, G. Zhang, C. Zhou, T. Liu, S. Lu, and L. Pan, "TransPillars: Coarse-to-Fine aggregation for multi-frame 3D object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4219–4228.

[28] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-lane: LiDAR lane dataset and benchmark for urban roads and highways," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4449–4458.

[29] F. Yan et al., "ONCE-3DLanes: Building monocular 3D lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17122–17131.

[30] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 642–657, May 2019.

[31] M. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 8748–8757.

[32] T. Zheng et al., "RESA: Recurrent feature-shift aggregator for lane detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3547–3554.

[33] P. Martinek, G. Pucea, Q. Rao, and U. Sivalingam, "LiDAR-based deep neural network for reference lane generation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 89–94.

[34] Z. Qin, W. Huanyu, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 276–291.

[35] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3773–3782.

[36] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8949–8958, Jul. 2022.

[37] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: end-to-end traffic line detection with line proposal unit," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 248–258, Jan. 2020.

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

[39] L. Tabelini, R. Berriel, T. M. Paix ao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 294–302.

[40] S. Huang et al., "Anchor3DLane: Learning to regress 3D anchors for monocular 3D lane detection," 2023, *arXiv:2301.02371*.

[41] L. Tabelini, R. Berriel, T. M. Paix ao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "PolyLaneNet: Lane estimation via deep polynomial regression," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6150–6156.

[42] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3694–3702.

[43] Y. Bai, Z. Chen, Z. Fu, L. Peng, P. Liang, and E. Cheng, "CurveFormer: 3D lane detection by curve propagation with curve queries and attention," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 7062–7068.

[44] H. Zhou, H. Zhou, J. Chang, T. Lu, and J. Ma, "3D lane detection from front or surround-view using joint-modeling & matching," *IEEE Trans. Intell. Vehicles*, early access, May 29, 2024, doi: 10.1109/TIV.2024.3406867.

[45] Z. Yang, Y. Zhou, Z. Chen, and J. Ngiam, "3D-MAN: 3D multi-frame attention network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1863–1872.

[46] Z. Qin, J. Chen, C. Chen, X. Chen, and X. Li, "Unifusion: Unified multi-view fusion transformer for spatial–temporal representation in bird's-eye-view," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Jun. 2023, pp. 8690–8699.

[47] S. Li et al., "DTCLMapper: Dual temporal consistent learning for vectorized HD map construction," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 21672–21686, Dec. 2024.

[48] T. Yuan, Y. Liu, Y. Wang, Y. Wang, and H. Zhao, "StreamMapNet: Streaming mapping network for vectorized online HD map construction," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 7356–7365.

[49] Z. Li et al., "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. ECCV*, vol. 13669, 2022, pp. 1–18.

[50] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[52] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, pp. 83–97, Mar. 1955.

[53] X. Yang et al., "Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Jan. 2021, pp. 18381–18394.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2017, pp. 1–11.

[56] Y. Wiseman, "Real-time monitoring of traffic congestions," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2017, pp. 501–505.

Ruixin Liu received the B.S. degree in computer science and technology from Tianjin University, Tianjin, China, in 2019. She is currently pursuing the Ph.D. degree in control science and engineering with Xi'an Jiaotong University, Xi'an, China, under the supervision of Dr. Zejian Yuan. Her research interests include computer vision and deep learning.

Zejian Yuan (Member, IEEE) received the M.S. degree in electronic engineering from Xi'an University of Technology, Xi'an, China, in 1999, and the Ph.D. degree in pattern recognition and intelligent systems from Xi'an Jiaotong University, Xi'an, in 2003. He is currently a Professor with the College of Artificial Intelligence, Xi'an Jiaotong University. His research interests include image processing, pattern recognition, and machine learning in computer vision. He is a member of Chinese Association of Robotics.