# LFP: Efficient and Accurate End-to-End Lane-Level Planning via Camera-LiDAR Fusion

Guoliang You[1], Xiaomeng Chu[1], Yifan Duan[1], Xingchen Li[1], Sha Zhang[1,3], Haojie Ren[1], Jianmin Ji[1], Yanyong Zhang[2], *Fellow, IEEE*

***Abstract*—Multi-modal systems enhance performance in autonomous driving but face inefficiencies due to indiscriminate processing within each modality. Additionally, the independent feature learning of each modality lacks interaction, which results in extracted features that do not possess complementary characteristics. This issue increases the cost of fusing redundant information across modalities. To address these challenges, we propose targeting driving-relevant lane elements, effectively reducing the volume of LiDAR features while preserving critical information. This approach enhances interaction at the lane level between the image and LiDAR branches, allowing for the extraction and fusion of their respective advantageous features. Building upon the camera-only framework PHP [1], we introduce the Lane-level camera-LiDAR Fusion Planning (LFP) method, which balances efficiency with performance by using lanes as the unit for sensor fusion. Specifically, we designed three novel modules to enhance efficiency and performance. For efficiency, we propose an image-guided coarse lane prior generation module that forecasts the region of interest (ROI) for lanes and assigns a confidence score, guiding LiDAR processing and fusion. The LiDAR feature extraction modules leverage lane-aware priors from the image branch to guide sparse sampling for pillar features, retaining essential features. For performance, the lane-level cross-modal query integration and feature enhancement module uses a confidence score from ROI to combine low-confidence image queries with LiDAR queries, extracting complementary depth features. These features then enhance the corresponding low-confidence image features, compensating for the lack of depth. Experiments on the Carla benchmarks show that our method achieves state-of-the-art performance in both driving score and infraction score, with a maximum improvement of 12.8% and 11.0% over existing algorithms, respectively, while maintaining a high frame rate of 19.27 FPS.**

## I. INTRODUCTION

AUTONOMOUS driving plays a vital role in improving the efficiency and safety of transportation systems [2], intersecting the fields of computer vision and robotics [3]–[5]. A key challenge lies in devising efficient and accurate planning [6]. Autonomous driving planning algorithms can be broadly categorized into two types: rule-based planning [7]–[10] and end-to-end learning-based planning [11]–[16]. Traditional rule-based planning leverages heavy algorithms, such as BEV-LaneDet [8] and BEVFormer [7], to detect lanes and position vehicles and pedestrians. Multi-modal algorithms [17], [18] are also employed to augment perception capabilities. These are typically followed by optimization techniques like the rapidly exploring random tree (RRT) [9] to identify safe paths with high computational cost. End-to-end learning-based planning algorithms, such as ST-P3 [14] and UniAD [15], leverage camera data and deep learning to streamline the planning process, reducing computational overhead [11]–[13]. Another class of camera-based methods, including PHP [1], attempts to further improve efficiency by selectively processing driving-relevant elements through a lane-level perception and planning method. Despite this improvement, pure vision systems still struggle with depth perception, which impairs environmental understanding and planning performance. To overcome these, methods such as TransFuser [19] and Interfuser [20] incorporate LiDAR's depth data into the end-to-end planning framework through attention mechanisms, significantly boosting perception and planning accuracy. For better fusion, the bird's eye view (BEV) offers an advantageous perspective for multi-modal fusion, providing a holistic view that streamlines the integration of camera and LiDAR data, as demonstrated by Think Twice [21]. However, indiscriminate processing of the vast amount of multimodal BEV information leads to high computational costs, as not all environmental elements are critical to planning. Moreover, the lack of interactive feature extraction between different modalities does not fully exploit their complementary strengths.

Given these challenges, the question arises: Can we achieve efficient and complementary feature extraction and fusion between LiDAR and camera? In response, we introduce LFP, a novel lane-level camera-LiDAR fusion method that extends PHP [1] for multi-modal fusion. Using lane priors from the

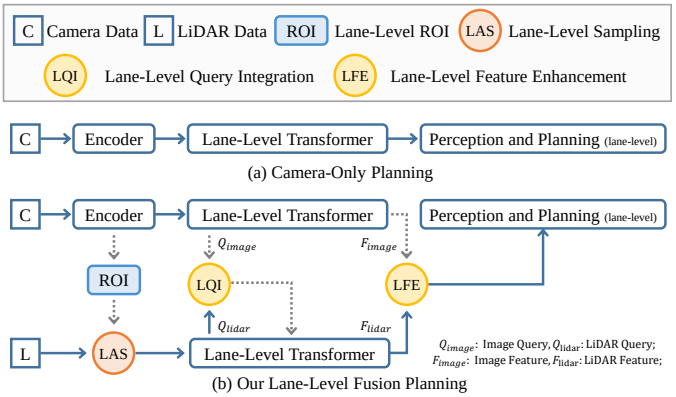


Fig. 1. An illustration comparing (a) the Camera-Only end-to-end planning with (b) our proposed lane-level Camera-LiDAR fusion end-to-end planning, where we utilize the geometric lane priors from images to guide the LiDAR branch in efficiently extracting depth features that the image branch lacks.

[1] School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei 230026, China
[2] School of Artificial Intelligence and Data Science, University of Science and Technology of China (USTC), Hefei 230026, China
[3] Shanghai AI Laboratory, Shanghai 200000, China
† Corresponding author. yanyongz, jianmin@ustc.edu.cn

camera branch, LFP minimizes essential LiDAR features, retaining only key information while improving the interaction between LiDAR and camera data, thus ensuring the extraction of complementary features.

Specifically, in LFP, the image branch, termed the image-guided coarse lane prior generation, processes images from multiple cameras to distill features. These features are submitted to a transformer that extracts lane-level image features. In parallel, the coarse lane detection module leverages the semantic richness of images to delineate lane geometric priors and quantify their confidence. The LiDAR branch, termed lane-level LiDAR feature extraction, leverages the Lane ROI for a sparse sampling of LiDAR pillars, focusing on critical lane regions. The fusion branch, guided by lane priors and termed lane-level cross-modal query integration and feature enhancement, first integrates the lane-level queries from both the image and LiDAR branches, refining the LiDAR branch's transformer retrieval of lane-level features. Then, through a weighted integration strategy, it maximizes the strengths of each modality, thereby enhancing the system's overall perception and planning capabilities. Finally, the fused features are utilized in the lane-level planning module to produce accurate perception and planning results.

This work shows that a lane-level camera-LiDAR fusion method, guided by lane priors from images, efficiently targets key driving elements within the LiDAR point cloud, avoiding unnecessary computations. Meanwhile, it promotes lane-level interaction to extract and fuse the most advantageous features from both the camera and LiDAR branches, enhancing the precision of planning.

In summary, our main contributions are as follows:

- We introduce LFP, a lane-level camera-LiDAR fusion planning algorithm that efficiently integrates LiDAR, enhancing feature depth and semantic richness via camera-LiDAR interaction, boosting planning performance.
- We developed a method with a coarse prior extraction algorithm that generates lane-level priors, enabling sparse LiDAR feature sampling and guiding the extraction and fusion of complementary features between camera and LiDAR branches at both the query and feature levels.
- We conducted experiments on the Carla benchmarks, demonstrating that LFP achieved state-of-the-art performance in both driving score and infraction score, with a maximum improvement of 12.8% and 11.0%, and maintained high computational efficiency at up to 19.27 FPS.
- To contribute to the community, the code will be open-sourced at https://github.com/ColorfulSS/LFP.

## II. RELATED WORK

### A. Vision-Based End-to-End Autonomous Driving

Vision-based End-to-end learning methods, such as those presented by Amini et al. [22] and Bojarski et al. [23], have made strides in mapping raw visual inputs directly to policies. In addition, researchers have explored various strategies. Codevilla et al. [24] and Chen et al. [25] have delved into the nuances of behavior cloning, identifying its limitations, and proposing novel techniques to enhance the learning process. The incorporation of reinforcement learning, as shown by Zhang et al. [26], has further enriched training regimens. Moreover, the advent of spatial-temporal feature learning, as demonstrated by Hu et al. [27], and the innovative vectorized scene representation by Jiang et al. [28], has paved the way for more effective planning. For comprehensive driving understanding, these methods may still underperform when relying solely on visual data. Systems that integrate additional sensors such as LiDAR, which provide critical depth information, generally perform better.

### B. Multi-Modal End-to-End Autonomous Driving

The advancement of autonomous driving has been significantly propelled by the integration of multi-modal data. Recognizing the value of additional information, Chitta et al. [29] and Chen et al. [30] leveraged camera and LiDAR data to achieve multi-modal fusion, showcasing enhanced planning performance. By emphasizing multi-modal temporal and global reasoning in driving scenarios, as showcased in their previous work [31], Shao et al. [32] have further demonstrated the capability of using language instructions and multi-modal sensor data as input to generate control signals for driving. Additionally, interpretability and safety in autonomous systems have been addressed by Shao et al. [33] through the development of transformer-based sensor fusion models. Jia et al. [34] proposed a novel multi-modal method that decouples perception and planning, allowing for a more flexible system design. Furthermore, Jia et al. proposed ThinkTwice [21], which focused on developing scalable decoders for multi-modal end-to-end planning, and DriveMLM by Wang et al. [35], which aligned multi-modal large language models with behavioral planning states. Collectively, these works show how multi-modal data fusion benefits autonomous driving.

## III. METHODOLOGY

### A. Preliminary

In LFP, we continue to employ the double-edge data structure from the PHP framework [1] and further propose a lane-level camera-LiDAR fusion planning method. PHP is a pure vision-based method that achieves lane-level planning by leveraging multi-view camera data. PHP defines $N_d$ double-edge data ($l_d^i$) that incorporate lane-level and point-level traffic information data to describe the environment $L$:

$$\begin{aligned} L &= \left\{l_d^i\right\}_{i=0}^{\mathrm{N}_d}, \\ l_d^i &= \left(edge_l^i, edge_r^i, int^i, dir^i\right), \\ int^i &= \{0 \ or \ 1\}, dir^i = \{0 \ or \ 1\}, \end{aligned} \tag{1}$$

Here, $l_d^i$ includes the lane's left and right $edge_i$, along with its lane-level intersection $int_i$ and direction $dir_i$ attributes. Within the double-edge, $int_i$ and $dir_i$ indicate whether a lane is an intersection and if the lane's direction is aligned with the direction of the ego vehicle's travel, respectively. $edge^i$

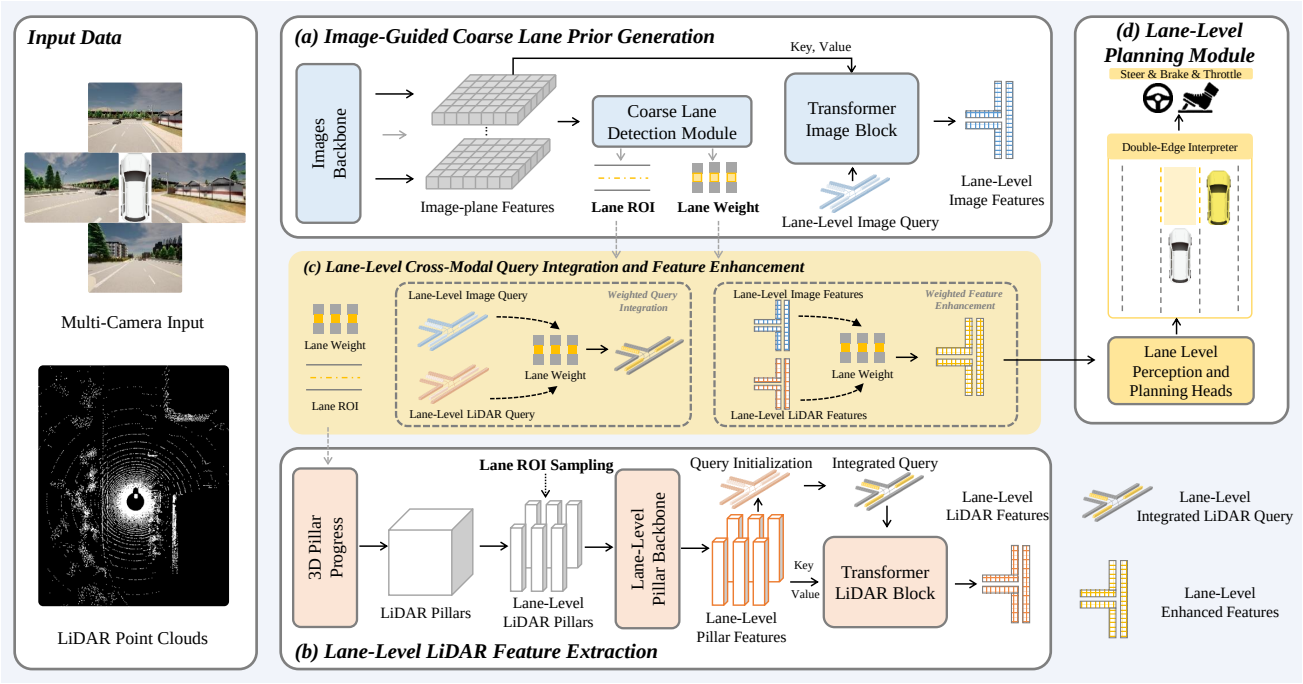


Fig. 2. The LFP integrates image and LiDAR through four modules: (a) The image-guided coarse lane prior generation module, which extracts lane-level image features and generates coarse lane priors (Lane ROI and Lane Weight); (b) The lane-level LiDAR feature extraction module, which performs pillar-based sampling guided by lane priors to focus on lane areas and extracts lane-level LiDAR features; (c) The lane-level cross-modal query integration and feature enhancement module, which integrates queries and features from both image and LiDAR at the query and feature levels; (d) The lane-level planning module, which processes the lane-level enhanced features, outputs lane-level perception and planning results, and converts them into vehicle control signals.

encompasses points in BEV and attributes for corresponding points, including occupancy and planning attributes that indicate the point level. Each $edge$ consists of $\frac{N_p}{2}$ elements, defined as:

$$\begin{aligned} edge^i &= \{point^j, occ^j, plan^j\}_{j=0}^{\frac{N_p}{2}-1}, \\ point^j &= \{x, y\}, \text{occ}^j = \{0 \ or \ 1\}, plan^j = \{0 \ or \ 1\}, \end{aligned} \tag{2}$$

where $occ^j$ and $plan^j$ indicate whether the lane is occupied by agents (e.g., pedestrians, vehicles) and whether it is selected for planning. Ultimately, we utilize the lane-level planning module (SecIII-E) to predict this structure, enabling the output of lane-level perception and planning results.

### B. Image-Guided Coarse Lane Prior Generation

This module extracts coarse lane priors from multi-view images by leveraging lane-level image queries to capture lane-level features from the environment, as illustrated in Figure 2(a). It comprises two main components: the transformer image block and the coarse lane detection module.

For the transformer image block, each image input $\mathbf{I} \in \mathbb{R}^{3\times H'\times W'}$ , we employ a ResNet-50 [36] to extracting features $\mathbf{f} \in \mathbb{R}^{C\times H\times W}$. The values for $C$, $H$, and $W$ are defined as $C = 256$, $H = \frac{H'}{32}$, and $W = \frac{W'}{32}$. The dimension of the transformer hidden layer is $E$. For each feature $\mathbf{f}$, we applies a $1 \times 1$ convolution to generate a lower-channel feature $\mathbf{z} \in \mathbb{R}^{E\times H\times W}$. Next, we simplify the spatial dimensions of $\mathbf{z}$ into a sequence, forming $E \times HW$ tokens. A fixed sinusoidal positional encoding $\mathbf{e} \in \mathbb{R}^{E\times HW}$ is then added to each token to preserve positional information within each sensor input:

$$\mathbf{v}_i^{(x,y)} = \mathbf{z}_i^{(x,y)} + \mathbf{e}^{(x,y)}, \tag{3}$$

where $\mathbf{z}_i$ represents the tokens extracted from the $i-th$ view, and $x$ and $y$ denote the token's coordinate index in that sensor. Subsequently, we concatenate $\mathbf{v}_i$ from all images and pass them through a transformer image block comprising $K$ standard transformer encoder and decoder layers. The transformer image block leverages $N_d$ lane-level image queries $\mathbf{q_{lane-level}} \in \mathbb{R}^{E\times N_p}$ to process and extract lane-level features $\mathbf{f_{image}}$, focusing on driving-relevant areas.

For the coarse lane detection module, we detect the geometric information of lane ROI in the image and quantify the confidence of these regions by introducing a soft label [37] mechanism that evaluates the model's capability in detecting depth within these regions. Firstly, this module processes the feature vectors $\mathbf{v}_i$ to extract a set of $N_d \times N_p \times 2$ dimensional coarse lane ROI ($\mathbf{R_{lane}}$), which represent the rough geometric information of the lane boundary points ($point^j$) within the double-edge. Concurrently, it generates lane weights ($\mathbf{W_{lane}}$) with a dimension of $N_d \times N_p$, which quantifies the confidence of these lane priors. This process is accomplished through a sequence of linear layers followed by ReLU and Softmax activations. The soft labels $y_j$ are generated based on the deviation between the predicted lane boundary points $\mathbf{P}_{\text{pred}}$ (where each point $\mathbf{P}_{\text{pred},j}$ corresponds to $point^j$) and the ground truth (GT) lane boundary points $\mathbf{P}_{\text{gt}}$. Specifically,

for each predicted point $\mathbf{P}_{\text{pred},j}$, we calculate its Euclidean distance $d_j$ to the corresponding GT point $\mathbf{P}_{\text{gt},j}$:

$$d_j = \|\mathbf{P}_{\text{pred},j} - \mathbf{P}_{\text{gt},j}\|_2, \tag{4}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Based on the distance $d_j$, we assign a soft label $\hat{y}_j$ to each point:

$$\hat{y}_j = \max\left(1 - \frac{d_j}{\tau}, 0\right), \tag{5}$$

where $\tau$ is the BEV perception range, which is set to the diagonal length of the perception area to ensure that the soft label $\hat{y}_j$ remains within the range $[0, 1]$. This mechanism enables the model to evaluate depth information more effectively by learning from accurate and uncertain predictions.

### C. Lane-Level LiDAR Feature Extraction

This module efficiently extracts driving-relevant and lane-level features from LiDAR by leveraging image-based lane priors to focus on critical depth data, minimizing computational redundancy, as shown in Figure 2(b).

The process begins with the pillar processing of the LiDAR points $\mathbf{P}_{\mathbf{lidar}} \in \mathbb{R}^{N\times 3}$, where $N$ represents the number of points. By processing the LiDAR point within pillars, where the height of the pillars is adjusted to match the actual height of the LiDAR points, we obtain a LiDAR pillar set $\mathbf{V}_{\mathbf{point}}$. However, direct feature encoding from these pillars still includes irrelevant data for driving tasks, such as trees and buildings. To retain the key depth information that the image branch lacks and reduce computational redundancy, we employ a lane-level sampling operation based on the image branch's lane ROI priors ($\mathbf{R}_{\mathbf{lane}}$). Utilizing the geometric positions of the lane ROI priors, we identify and retain the closest pillars to these priors, resulting in a set of lane-level pillars $\mathbf{V}_{\mathbf{lane}}$ with dimensions $N_d \times N_p \times C$, where C denotes the dimension of each pillar. This selective retention ensures that only the pillars critical to driving are preserved, allowing the LiDAR branch to focus on the most relevant data.

Subsequently, the lane-level pillar backbone processes the lane-level pillars $\mathbf{V}_{\mathbf{lane}}$ to obtain the lane-level pillar features $\mathbf{f}_{\mathbf{lane}}$. These features are then used to initialize the lane-level LiDAR queries $\mathbf{q}_{\mathbf{lidar}}$ in the transformer LiDAR block, which have dimensions $N_d \times N_p \times E$. The transformer LiDAR block then leverages these queries to perform feature extraction through $K$ layers of multi-head self-attention. The primary objective is to refine the lane-level pillar features into lane-level LiDAR features $\mathbf{f}_{\mathbf{lidar}}$. During this stage, we also prepare for the cross-modality query integration, which will be detailed in SecIII-D. This integration is anticipated to incorporate the image branch's query $\mathbf{q}_{\mathbf{image}}$ into the LiDAR query $\mathbf{q}_{\mathbf{lidar}}$, thereby enabling the LiDAR branch to target on the lane regions of interest as identified by the image branch.

### D. Lane-Level Cross-Modal Query Integration and Feature Enhancement

This section details the cross-modal query integration, harnessing image branch semantics to guide LiDAR depth feature extraction, complementing the image's depth limitations. It also outlines a feature enhancement strategy that merges branch strengths for efficient lane-level feature fusion, optimizing performance and computational efficiency, as in Figure 2(c). Specifically, the image branch, enriched with semantic information but limited in depth, excels at concentrating feature queries around the lanes. In contrast, the LiDAR branch, proficient in capturing depth information, may struggle to identify lanes due to the lack of semantic cues. To address this, we propose a weighted integration of the image branch's query $\mathbf{q}_{\mathbf{image}}$ and the LiDAR branch's initialized query $\mathbf{q}_{\mathbf{lidar}}$, with weights determined by the lane weights $\mathbf{W}_{\mathbf{lane}}$ associated with the lane ROI. This integration results in a lane-level LiDAR integrated query, $\mathbf{q}_{\mathbf{integrated}}$, adept at querying the lane-level pillar features $\mathbf{f}_{\mathbf{lane}}$ to extract the critical depth features that the image lacks. The integration enhances the focus of LiDAR queries on areas where image depth cues are insufficient, directing the LiDAR branch to supplement the image depth features with precise depth information. $\mathbf{q}_{\mathbf{integrated}}$ is defined as:

$$\mathbf{q}_{\text{integrated}} = (1-\alpha)\cdot \mathbf{q}_{\text{image}} + \alpha \cdot \mathbf{q}_{\text{lidar}}, \tag{6}$$

where $\alpha$ represents the weight derived from the lane weight. After both branches have processed their sensor data and obtained lane-level features, a fusion of these features is necessary to leverage the semantic prowess of the image branch and the depth acuity of the LiDAR branch. We achieve this by employing a similar weighted fusion approach, combining the lane-level features from the image branch $\mathbf{f}_{\mathbf{image}}$ and the LiDAR branch $\mathbf{f}_{\mathbf{lidar}}$ with weights $\mathbf{W}_{\mathbf{lane}}$, leading to an enriched set of lane-level enhanced features $\mathbf{f}_{\mathbf{enhanced}}$:

$$\mathbf{f}_{\text{enhanced}} = \beta \cdot \mathbf{f}_{\text{image}} + (1-\beta)\cdot \mathbf{f}_{\text{lidar}}, \tag{7}$$

where $\beta$ is the weight that balances contributions from both branches, derived from the lane weight.

### E. Lane-Level Planning Module

This module predicts lane-level perception and planning tasks using features $\mathbf{f}_{\text{enhanced}}$ , including a lane-level perception and planning head for double-edge data structure and a double-edge interpreter for safe planning results, as in Figure 2(d).

*1) Lane-Level Perception and Planning Heads:* LFP integrates a series of prediction heads for the double-edge. A regression head predicts boundary points $point^j$, and four classification heads forecast attributes of intersection $int_i$, direction $dir_i$, occupancy $occ^j$, and planning $plan^j$ within the traffic scenario. Following previous pure vision-based lane-level autonomous driving works such as PHP, our approach further enhances the perception and planning capabilities by leveraging integrated geometric and semantic features.

*2) Lane-Level Double-Edge Interpreter:* The double-edge interpreter translates the rich geometric and attributive lane-level information from the double-edge into an executable path by a controller. For path generation, we select edge

points marked by a planning attribute value of '1' (indicating suitability for planning) to construct the path:

$$Path = \bigcup_{j=1}^{N_d \times \frac{N_p}{2}} \left\{ \frac{point_l^j + point_r^j}{2} \middle| plan_l^j, plan_r^j = 1 \right\}. \quad (8)$$

By leveraging the image branch's strength in traffic signal recognition, we incorporate speed and traffic signal queries to predict velocity and traffic conditions. These predictions, including speed and path, are fed into the interpreter and then directly utilized by controllers [38] to convert them into control signals, such as brake, throttle, and steering angle, achieving closed-loop vehicle control.

### F. Loss Function

LFP predicts coarse priors and executes perception and planning, including $L_{roi}$ for coarse lane prior prediction, $L_{edg}$ for double-edge regression, $L_{int}$ and $L_{dir}$ for intersection and direction respectively, $L_{occ}$ and $L_{plan}$ for occupancy and planning respectively. Additionally, $L_{spd}$ and $L_{sig}$ are used for speed and traffic signals. Formulated as:

$$\begin{aligned} Loss = \gamma L_{roi} + \delta L_{edg} + \epsilon L_{dir} + \varepsilon L_{occ} + \\ \zeta L_{plan} + \eta L_{int} + \theta L_{spd} + \iota L_{sig}, \end{aligned} \quad (9)$$

where, in training, $\gamma$, $\delta$, $\epsilon$, $\varepsilon$, $\zeta$, $\eta$, $\theta$ and $\iota$ are set to 3:2:5:1:3:4:1:0.1. The coarse prior $L_{roi}$ is formulated as:

$$L_{roi} = \frac{1}{N_{gt}} \sum_{i=0}^{N_{gt}-1} \sum_{j=0}^{\frac{N_p}{2}-1} \left\{ \left| y_{ij}^l - \hat{y}_{ij}^l \right| + \left| y_{ij}^r - \hat{y}_{ij}^r \right| \right\}, \quad (10)$$

where $N_{gt}$ denotes the number of ground truth lanes. Here, $y_{ij}^l$ and $y_{ij}^r$ are the predicted confidence scores, and $\hat{y}_{ij}^l$ and $\hat{y}_{ij}^r$ are the ground truth confidence scores for the left and right lane boundaries at the $j$-th point of the $i$-th ground truth lane, respectively. For perception and planning, $L_{edg}$ utilizes the Manhattan distance for point regression, $L_{int}$, $L_{dir}$, and $L_{occ}$ employ Focal Loss [39], while $L_{spd}$ uses SmoothL1Loss [40], $L_{sig}$ is Cross-Entropy Loss, $L_{edg}$ and $L_{plan}$ can be formulated as:

$$L_{edg} = \frac{1}{N_{gt}} \sum_{i=0}^{N_{gt}-1} \sum_{j=0}^{\frac{N_p}{2}-1} \left\{ \left| pred_{ij}^l - gt_{ij}^l \right| + \left| pred_{ij}^r - gt_{ij}^r \right| \right\}, \quad (11)$$

$$L_{plan} = \sum_{i=0}^{N_{gt}-1} \sum_{j=0}^{\frac{N_p}{2}-1} \left\{ \frac{\left(\rho \cdot \left(1 - e^{-CE(pred_{ij}, gt_{ij})}\right)\right)^2 \cdot CE(pred_{ij}, gt_{ij})}{D_{ij}^{p2t}} \right\}, \quad (12)$$

where $D_{p2t}$ represents the distance from an edge point into the target vector, serving as weights to emphasize planning features near target points, $CE$ refers to the cross-entropy loss, and $\rho$ is set to 0.25.

TABLE I
PERFORMANCE COMPARISON ON CARLA TOWN05 LONG.

| Method (Year) | Modality | DS ↑ | RC ↑ | IS ↑ |
|---|---|---|---|---|
| CILRS'19 [41] | C | 7.8±0.3 | 10.3±0.0 | 0.75±0.05 |
| LBC'20 [42] | C | 12.3±2.0 | 31.9±2.2 | 0.66±0.02 |
| NEAT'21 [16] | C | 37.7±3.6 | 62.1±4.7 | - |
| Roach'21 [43] | C | 41.6±1.8 | 96.4±2.1 | 0.43±0.03 |
| WOR'21 [44] | C | 44.8±3.7 | 82.4±5.0 | - |
| ST-P3'22 [14] | C | 11.5±- | 83.2±- | - |
| Transfuser'22 [19] | C+L | 31.0±3.6 | 47.5±5.3 | 0.77±0.04 |
| Interfuser'22 [20] | C+L | 68.3±1.9 | 95.0±2.9 | - |
| VAD'23 [45] | C | 30.3±- | 75.2±- | - |
| ThinkTwice'23 [21] | C+L | 70.9±3.4 | 95.5±2.6 | 0.75±0.05 |
| DriveAdapter'23 [34] | C+L | 71.9±- | 97.3±- | 0.74±- |
| DriveMLM'23 [35] | C+L | 76.1±- | 98.1±- | 0.78±- |
| ReasonNet'23 [31] | C+L | 73.2±1.9 | 95.9±2.3 | 0.76±0.03 |
| LAW'24 [46] | C | 70.1±2.6 | 97.8±0.9 | 0.72±0.03 |
| PHP'24 [1] | C | 78.3±1.7 | 96.2±1.3 | 0.81±0.02 |
| **LFP** (*ours*) | C+L | **91.1±0.9** | **99.6±0.3** | **0.92±0.01** |

TABLE II
PERFORMANCE COMPARISON ON CARLA TOWN05 SHORT.

| Method (Year) | Modality | DS ↑ | RC ↑ |
|---|---|---|---|
| CILRS'19 [41] | C | 7.5±2.5 | 13.4±1.1 |
| LBC'20 [42] | C | 31.0±4.2 | 55.0±5.1 |
| NEAT'21 [16] | C | 58.7±4.1 | 77.3±4.9 |
| Roach'21 [43] | C | 65.3±3.6 | 88.2±5.2 |
| WOR'21 [44] | C | 64.8±5.5 | 87.5±4.7 |
| Transfuser'22 [19] | C+L | 54.5±4.3 | 78.4±3.8 |
| ST-P3'22 [14] | C | 55.1±- | 86.7±- |
| Interfuser'22 [20] | C+L | 95.0±2.0 | 95.2±2.6 |
| VAD'23 [45] | C | 64.3±- | 87.3±- |
| ReasonNet'23 [31] | C+L | 95.7±1.9 | 96.2±3.2 |
| LeapAD'24 [47] | C | 75.7±1.4 | 92.1±1.44 |
| PHP'24 [1] | C | 92.5±1.5 | 97.0±1.2 |
| **LFP** (*ours*) | C+L | **96.7±1.9** | **98.3±1.2** |

## IV. EXPERIMENTS

### A. Dataset and Metrics

Using the autonomous driving environment Carla [48], we gather 126K frames from diverse scenarios across 8 maps and 13 weathers. The data are collected at 2Hz with vehicles equipped with four cameras, a LiDAR, an IMU, and a GPS. And, we annotate 3D edge points with attributes including intersection, direction, occupancy, and planning. We adopt a rigorous benchmark setup: the model is trained on data from Town01∼04, 06∼07, and 10, and tested on Town05. Evaluation metrics encompass Driving Score (DS), Route Completion (RC), and Infraction Score (IS). RC signifies the proportion of the route navigated by the agent, while IS indicates penalties incurred from accidents. Multiplying RC by IS yields the final metric DS, which evaluates the driving performance of our method on a route.

### B. Performance on Carla Benchmark

In this section, we employ the Town05 Long and Town05 Short benchmarks for closed-loop evaluation to demonstrate how LFP leverages the lane-level Camera-LiDAR fusion method to achieve superior planning performance while maintaining high computational efficiency. The Town05 Long

TABLE III
COMPARATIVE ANALYSIS OF EFFICIENCY.

| Method (Year) | Modality | Latency (ms) ↓ | FPS ↑ |
|---|---|---|---|
| CILRS'21 [41] | C | 152.88 | 6.54 |
| LBC'20 [42] | C | 91.96 | 10.87 |
| NEAT'21 [16] | C | 85.08 | 11.75 |
| ST-P3'22 [14] | C | 476.74 | 2.1 |
| TransFuser'22 [19] | C+L | 171.93 | 5.82 |
| VAD'23 [45] | C | 59.50 | 16.81 |
| ThinkTwice'23 [19] | C+L | 262.77 | 3.81 |
| PHP'24 [1] | C | 44.30 | 22.57 |
| **LFP (ours)** | C+L | 51.90 | 19.27 |

TABLE IV
COMPARATIVE ANALYSIS OF LIDAR FEATURE COUNTS.

| Method | Resolution | LiDAR Features |
|---|---|---|
| LiDAR Voxel | [0.5, 0.5, 0.5] | 4364 |
| LiDAR Pillar | [0.5, 0.5, 8] | 3446 |
| Lane-Level LiDAR Pillar | [0.5, 0.5, 8] | **600** |

benchmark comprises 10 routes, with each spanning approximately 1km. Conversely, the Town05 Short benchmark includes 32 routes, each measuring 70m. These benchmarks collectively assess the model's overall performance, demonstrating state-of-the-art capabilities in end-to-end planning.

*1) Driving Score:* The driving score (DS) comprehensively assesses the autonomous driving system's performance, combining route completion and infraction scores. In the Town05 Long and Short benchmarks, LFP outperforms all other algorithms, including both camera-only and camera-LiDAR fusion methods, by achieving the highest driving score. Specifically, in the Town05 long benchmarks, our system achieves a driving score improvement of 12.8% (Long) and 1.0% (Short), as detailed in Table I and II. These results underscore the effectiveness of LFP in enhancing safety and traffic regulation compliance.

*2) Route Completion:* The route completion (RC) is a critical metric for assessing an autonomous driving system's success in completing predetermined routes throughout the evaluation process, reflecting the system's ability to plan routes and accurately understand target points. In the Carla Town05 Long and Short benchmark, compared to other algorithms, our method shows a higher route completion rate in benchmark tests that included a diverse range of traffic scenarios. Specifically, our method achieves improvements of 1.5% and 1.3%, respectively, as shown in Table I and II. These results highlight the advanced capabilities of our method in route planning, as well as its robustness in adapting to various road conditions.

*3) Infraction Score:* The infraction score (IS) is a comprehensive metric used to evaluate systems performance, including avoiding collisions, adhering to traffic rules, and handling complex situations. LFP exhibits significant performance in Town05 Long, achieving a 11.0% improvement in infraction score (IS) as detailed in Table I, surpassing other algorithms.

*4) Computational Efficiency:* Fusing LiDAR and camera data can enhance system performance. However, improper integration of LiDAR data may reduce operational efficiency.

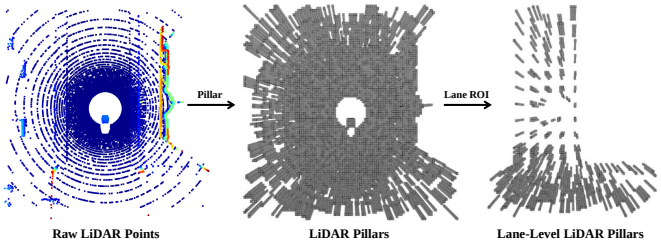


Fig. 3. Visualizing the Transformation from LiDAR to Lane-Level Pillars

Our method employs a lane-level fusion algorithm that extracts and integrates critical depth features required by the image branch, thereby enabling the system to process only essential depth data. This approach reduces computational redundancy in the LiDAR branch and minimizes the cost of multi-modal data fusion, achieving efficient data integration. As shown in Table III, our algorithm achieves a computational speedup of 19.27 FPS over other camera-LiDAR fusion methods, with the LiDAR branch's introduction adding only 7.6 ms overhead. Additionally, LeapAD, being built on LLMs, requires more computation time due to the inherent complexity of language models. Furthermore, Figure 3 visualizes the lane-level LiDAR pillar results, showing that depth data from only critical lane areas are extracted. Table IV quantitatively describes the features obtained by different LiDAR processing methods. After processing, the lane-level LiDAR pillars reduce the data volume by 7.27 times compared to Voxel and 5.74 times compared to Pillar methods. This results in higher processing efficiency for the LiDAR branch.

### C. Qualitative Results

In Figure 4, we illustrate our system's ability to hierarchically perceive the traffic environment and seamlessly integrate planning tasks at the lane level. The visualization includes intersection lanes marked in blue, direction lanes that indicate roads complying with traffic regulations marked in green, and occupancy lanes for roads unoccupied by traffic agents and adhering to directions marked in orange. The planning lane, highlighted in yellow, signifies the optimally chosen lane that ensures safety and leads to the target point.

### D. Ablation Studies

*1) Impact of Coarse Detection Lane Module:* This module utilizes image features to provide coarse lane ROI priors to the LiDAR branch, enabling it to learn critical depth information. To validate the module's influence, we conducted an experiment (Experiment A) where the learnable coarse lane priors were replaced with random lane ROI priors. Our findings indicate that random priors fail to direct the LiDAR branch toward the depth information of interest to the image branch, resulting in the fusion features lacking the desired depth cues. This deficiency leads to a degradation in performance. As shown in Table V, Experiment A's results demonstrate a decrease in driving score (DS), route completion (RC), and infraction score (IS) by 73.7%, 45.2%, and 46.0%, respectively,

TABLE V
ABLATION STUDY ON CARLA TOWN05 LONG. "CLM" DENOTES COARSE LANE DETECTION MODULE, "CQF" DENOTES LANE-LEVEL CROSS-MODAL QUERY INTEGRATION AND FEATURE ENHANCEMENT, AND "LFE" DENOTES LANE-LEVEL LIDAR FEATURE EXTRACTION.

| ID | CLM | CQF | LFE | Carla Town05 Long | | |
|---|---|---|---|---|---|---|
| – | Sec III-B | Sec III-D | Sec III-C | DS↑ | RC↑ | IS↑ |
| A | ✗ | ✔ | ✔ | 17.4 | 54.4 | 0.46 |
| B | ✔ | ✗ | ✔ | 50.2 | 87.0 | 0.57 |
| C | ✔ | ✔ | ✗ | 78.3 | 96.2 | 0.81 |
| **D** | ✔ | ✔ | ✔ | **91.1** | **99.6** | **0.92** |

compared to the original learnable module. This experiment underscores the importance of accurate lane ROI priors in enhancing the depth perception capabilities of the LiDAR branch and the overall performance of the LFP method.

*2) Impact of Lane-Level Cross-Modal Query Integration and Feature Enhancement:* This module is hinged on an adaptive weighting fusion mechanism, which we demonstrate as crucial for effective cross-modal fusion. In Experiment B, we removed the weighted query integration, opting to use only the initial LiDAR query. We also replaced the weighted feature enhancement with an equal-weight fusion of dual-branch features, which neutralized the depth information prioritization. This adjustment resulted in notable decreases in key metrics: driving score, route completion, and infraction score by 40.9%, 12.6%, and 35.0%, respectively, as shown in Table V. The results affirm that the adaptive weighting of lane priors is essential for the LiDAR branch to accurately extract and integrate depth features, thus enhancing the multi-modal system's fusion efficiency and overall performance.

*3) Impact of Lane-Level LiDAR Feature Extraction:* This module enhances the LFP by providing depth features complementary to the image branch. Removing this module reverts LFP to a camera-only approach (Experiment C), which, as shown in Table V, suffers from a lack of depth information, causing decreases in driving score, route completion, and infraction score by 12.8%, 3.4%, and 11.0%, respectively. This highlights the module's critical role in leveraging the complementary strengths of multi-modal data.

### *E. Deploying in Real-world Scenarios*

We deploy our autonomous driving platform to validate its performance in real-world environments. As shown in Figure 5 (a), the platform for the autonomous vehicle is built on an Agilex Hunter 2.0 chassis and is equipped with a 32-line 3D LiDAR for environmental perception. Additionally, the robot incorporates an RTX 3090 as its computing unit. The dimensions of the vehicle, length × width × height, are 0.95m × 0.75m × 1.45m. To provide effective multimodal data, we calibrate the camera and LiDAR extrinsically, undistort images using intrinsic parameters, and ensure precise temporal synchronization via hardware triggers. Our experiments demonstrate that the vehicle can accurately perceive lane-level elements and generate lane-level planning to complete navigation tasks, as shown in Figure 5 (b). More illustrations of the vehicle are shown in our demonstration video.

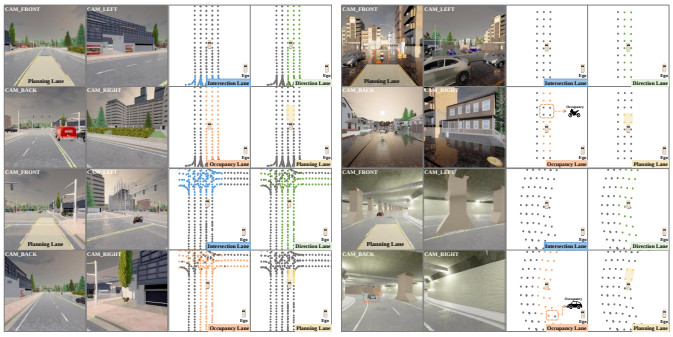

Fig. 4. Visualization of Lane-Level Perception and Planning in LFP.

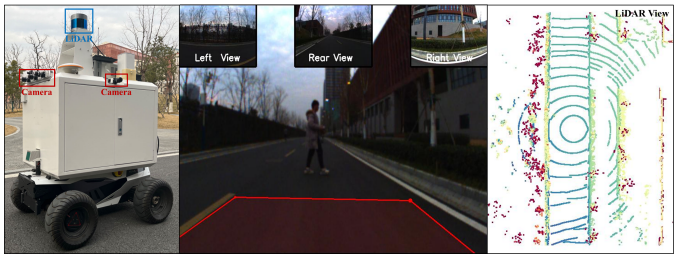


Fig. 5. Autonomous Driving Vehicle Platform and Real-World Scenario Case.

## V. CONCLUSIONS

In this paper, we presented the end-to-end lane-level camera-LiDAR fusion planning (LFP) method, which effectively balances performance with computational efficiency. LFP enhances efficiency by integrating image-guided lane prioritization with sparse LiDAR sampling, leveraging lane priors to minimize computational redundancy and focus on lane-relevant data. Additionally, LFP employs an efficient lane-level query integration and feature enhancement strategy, effectively merging semantic image information with LiDAR depth data to produce a comprehensive lane-level representation essential for planning. Experiments on the Carla Benchmark validate LFP's superior efficiency and performance, surpassing current algorithms. Our lane-level camera-LiDAR fusion strategy ensures operational excellence at 19.27 FPS, underscoring the benefits of our integrated approach.

## REFERENCES

[1] G. You, X. Chu, Y. Duan, W. Zhang, X. Li, S. Zhang, Y. Li, J. Ji, and Y. Zhang, "Perception helps planning: Facilitating multi-stage lane-level integration via double-edge structures," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2104–2111, 2025.

[2] C. Badue and R. Guidolini et al., "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[3] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," *Pattern Recognit.*, vol. 130, p. 108796, 2022.

[4] X. Li, Y. Xiao, B. Wang, H. Ren, Y. Zhang, and J. Ji, "Automatic targetless lidar–camera calibration: a survey," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9949–9987, 2023.

[5] M. U. Khan and S. A. A. Zaidi et al., "A comparative survey of lidar-slam and lidar based sensor technologies," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE, 2021, pp. 1–8.

[6] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[7] Z. Li, W. Wang, and et al., "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, ser. Lecture Notes in Computer Science, vol. 13669. Springer, 2022.

[8] R. Wang, J. Qin, K. Li, Y. Li, D. Cao, and J. Xu, "Bev-lanedet: An efficient 3d lane detection based on virtual camera via key-points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1002–1011.

[9] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[10] A. Tahirovic and M. Ferizbegovic, "Rapidly-exploring random vines (rrv) for motion planning in configuration spaces with narrow passages," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[11] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," *IEEE Transactions on Intelligent Vehicles*, 2023.

[12] M. Bojarski and D. Del Testa et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[13] W. Yu, J. Peng, Q. Qiu, H. Wang, L. Zhang, and J. Ji, "Pathrl: An end-to-end path generation method for collision avoidance via deep reinforcement learning," *CoRR*, vol. abs/2310.13295, 2023.

[14] S. Hu and L. Chen et al., "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 533–549.

[15] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[16] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 793–15 803.

[17] Z. Liu and H. Tang et al., "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[18] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.

[19] K. Chitta and A. Prakash et al., "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[20] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

[21] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li, "Think twice before driving: Towards scalable decoders for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 983–21 994.

[22] A. Amini, G. Rosman, S. Karaman, and D. Rus, "Variational end-to-end navigation and localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8958–8964.

[23] M. Bojarski and D. Del Testa et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[24] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[25] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[26] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 222–15 232.

[27] S. Hu and L. Chen et al., "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 533–549.

[28] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "VAD: vectorized scene representation for efficient autonomous driving," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 8306–8316. [Online]. Available: https://doi.org/10.1109/ICCV51070.2023.00766

[29] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[30] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 222–17 231.

[31] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 13 723–13 733.

[32] H. Shao, Y. Hu, L. Wang, G. Song, S. L. Waslander, Y. Liu, and H. Li, "Lmdrive: Closed-loop end-to-end driving with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 120–15 130.

[33] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

[34] X. Jia, Y. Gao, L. Chen, J. Yan, P. L. Liu, and H. Li, "Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7953–7963.

[35] W. Wang, J. Xie, C. Hu, H. Zou, J. Fan, W. Tong, Y. Wen, S. Wu, H. Deng, Z. Li, *et al.*, "Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving," *arXiv preprint arXiv:2312.09245*, 2023.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, June 27-30, 2016*. IEEE.

[37] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.

[38] N. Hung, F. Rego, J. Quintas, J. Cruz, M. Jacinto, D. Souto, A. Potes, L. Sebastiao, and A. Pascoal, "A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments," *Journal of Field Robotics*, vol. 40, no. 3, pp. 747–779, 2023.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[40] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[41] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[42] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[43] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 222–15 232.

[44] D. Chen, V. Koltun, and P. Krähenbühl, "Learning to drive from a world on rails," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 590–15 599.

[45] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "VAD: vectorized scene representation for efficient autonomous driving," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 8306–8316. [Online]. Available: https://doi.org/10.1109/ICCV51070.2023.00766

[46] Y. Li, L. Fan, J. He, Y. Wang, Y. Chen, Z. Zhang, and T. Tan, "Enhancing end-to-end autonomous driving with latent world model," *arXiv preprint arXiv:2406.08481*, 2024.

[47] J. Mei, Y. Ma, X. Yang, L. Wen, X. Cai, X. Li, D. Fu, B. Zhang, P. Cai, M. Dou, *et al.*, "Continuously learning, adapting, and improving: A dual-process approach to autonomous driving," *Advances in Neural Information Processing Systems*, 2024.

[48] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.