Appendix

We provide a detailed description of the K-Lane dataset and the development kits (devkits), and detailed structure of proposed LLDN-GFC with its CNN-based counterparts, in Section A, and B, respectively. In addition, Section C shows ablation study for the network hyper-parameters of the proposed LLDN-GFC (i.e., Proj28-GFC-T3), low computational alternative (i.e., Pillars-GFC-M5), and the counterparts. Furthermore, Section D shows qualitative lane detection results for K-Lane, and visualization of both heatmap of features and the attention score. Lastly, Section E shows the comparison between LLDN-GFC and heuristic lane detection methods.

A. K-Lane and Devkits

Section A contains technical details that may helps researchers in using the K-Lane datasets and devkits.

A.1. Details of K-Lane and Devkits

In this section, we present three additional details about the K-Lane: sequence distributions, compositions, and the criteria of driving conditions annotations of the dataset.

Sequence Distribution. K-Lane dataset consists of fifteen sequences that have different set of road conditions. The details of the sequences are shown in Table 3. For the test data, we provide additional driving conditions annotations on each frame (i.e., curve, occlusion, merging, and number of lanes) with annotation tool shown in Section A.2.

Conditions Criteria. To evaluate the LLDN performance depending on data characteristics, we provide 13 different categories of driving conditions as shown in Table 4. Examples of each condition are shown Fig. 1, and each frame can have two or more conditions, for example, day time and occlusion.

Dataset Composition. The K-lane is divided into fifteen directories, each representing a sequence. Each directory has one associated file that describe the driving condition of the frames in the sequence, and contains files for the collected point cloud data, BEV point cloud tensor (i.e., stacked pillars shown in Fig. 10), BEV label, front (camera) images, and calibration parameters, as shown in Table 5. Pedestrians' faces are blurred on the front images for privacy protection. Interface for pre-processing the files is provided in Section A.2.

A.2. Details of Development Kits

In addition to the K-Lane dataset, we also provide the devkits which can be used to expand the dataset, and to develop further LLDNs. The devkits are available to the public in the form of three programs: (1) TPC - Total Pipeline Code for training and evaluation, (2) GLT - Graphic User Interface (GUI)-based Labeling Tools, and (3) GDT – GUI-based development Tools for evaluation, visualization, and additional conditions annotations.

Sequence	Num. Frames	Location	Time
1	1708	Urban roads [city #1]	Night
2	803	Urban roads [city #2]	Day
3	549	Urban roads [univ. #1]	Day
4	1468	Urban roads [univ. #1]	Day
5	251	Urban roads [city #2]	Day
6	132	Urban roads [city #2]	Day
7	388	Urban roads [city #2]	Day
8	357	Urban roads [city #2]	Day
9	654	Urban roads [city #2]	Day
10	648	Urban roads [city #2]	Day
11	1337	Urban roads [city #2]	Night
12	370	Urban roads [city #2]	Night
13	2991	Highway [city #2 to city #3]	Day
14	1779	Highway [city #2 to city #3]	Night
15	1947	Highway [city #3 to city #2]	Night

Table 3. Sequences in K-Lane.

Total Pipeline Code. TPC is a complete neural network development code that supports pre-processing of the input data and label, train the network, and perform evaluation based on the F1-metric. TPC handles input and output as Python dictionaries and support modularization of the LLDN (BEV encoder, GFC, detection head), therefore, providing comprehensive and flexible support to developers.

GUI-based Labeling Tools. GLT provides an easy way to develop a labeled dataset for a Lidar and a front view camera, regardless of the Lidar and camera models. As shown in Fig. 8 (left), GLT provides an easy way for labeling by showing the intensity of point cloud in a BEV image. Fig. 8 (middle) shows a synchronized front camera image for easy labeling of point cloud, and Fig. 8 (right) shows the saved labeled point cloud.

GUI-based Development Tools. GDT is a GUI program used together with TPC. GDT provides visualization of inference results for each scene as point cloud or camera image with projected lanes (Fig. 9-b), high-accuracy calibration of camera and Lidar sensors with specific points of the lanes (Fig. 9-c), and annotation of each frame with set of buttons (Fig. 9-d).

B. Details of LLDNs

This section provides a detailed neural network structure of the LLDN-GFC proposed in Section 3.2 of the main paper and its counterparts, CNN-based LLDN.

B.1. Details of LLDN-GFC

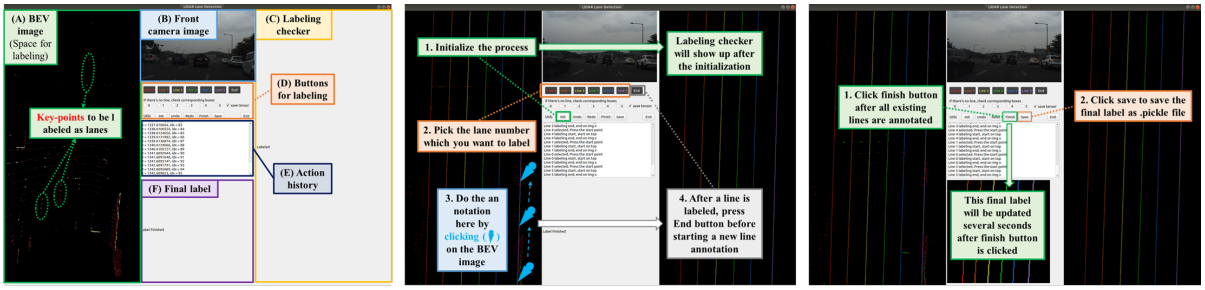
This section describes the sub-structure of the proposed

Conditions	Explanation	Num. Frames
Urban	Data acquired from city or university	8607
Highway	Data acquired on Highway	6775
Night	Data acquired at night (approximately 20:00-2:00)	7139
Daytime	Data acquired during the daytime (about 12:00-16:00)	8243
Normal	Data without curved or merging lanes (mostly straight lanes)	11065
Gentle Curve	Data with curved lanes whose radius of curvature is greater than 160 [m]	1804
Sharp Curve	Data with curved lanes whose radius of curvature is less than 160 [m]	1431
Merging	Data with a converging or diverging lane at the rightmost or leftmost lane	982
No Occlusion	Data without occluded lanes based on the lane label	9443
Occlusion 1	Data with one occluded lane based on the lane label	2660
Occlusion 2	Data with two occluded lanes based on the lane label	2112
Occlusion 3	Data with three occluded lanes based on the lane label	793
Occlusion 4-6	Data with four to six occluded lanes based on the lane label; Since there are few samples of data with five or six occluded lanes, they are integrated as a single condition (i.e., four to six occluded lanes).	374

Table 4. Condition details

Datum Type	Extension	Format	Comment
Point cloud	.pcd	Point cloud with 131072 points	Input to point projector and heuristic technique
BEV point cloud tensor	.pickle	$N_g \times N_c \times N_p$ size array	Input to pillar encoder
BEV label	.pickle	$H_{bev} \times (W_{bev} + 6)$ size array	Lane label including unlabeled lane per row (6 columns are for the possible row-wise detection-based approaches)
Front image	.png	RGB image	For annotation and visualization
Calibration parameters	.txt	Intrinsic and extrinsic parameters	For Lidar-camera projection
Condition	.txt	Condition (e.g., night and day)	For evaluation

Table 5. Dataset Composition



(a) (b) (c)

Figure 8. GUI-based Labeling Tool (GLT): (a) Overall components of GLT, (b) Labeling process of a point cloud, (c) Finalizing and saving the label.

baseline LLDN-GFC, first shown in Fig. 4. We divide the LLDN-GFC structure into three parts: the BEV encoder, the global feature corrector (GFC), and the detection head. The functions (1)∼(5) of Fig. 4 are equivalent to the functions (1)∼(5) of Fig. 10∼12. (e.g., (1) of Fig. 4 is equivalent to (1-1) and (1-2) of Fig. 10.)

BEV Encoder. BEV encoder projects 3D point cloud into a horizontal plane to produce 2D pseudo-BEV image. A large number of heuristic path planning algorithms, such as A* [4], RRT* [7], and End-to-End autonomous driving algorithms [1] require lane lines on 2D BEV images. The proposed LLDN-GFC variants use one of the two most common 2D BEV encoders, as shown in Fig. 10.

The primary 2D BEV encoder for the LLDN-GFC is the point projector [8, 12] that projects point clouds into xy-horizontal plane and produces pseudo-BEV images using CNN. In this case, three additional information (z, intensity, and reflectivity) other than x and y of the point cloud is used

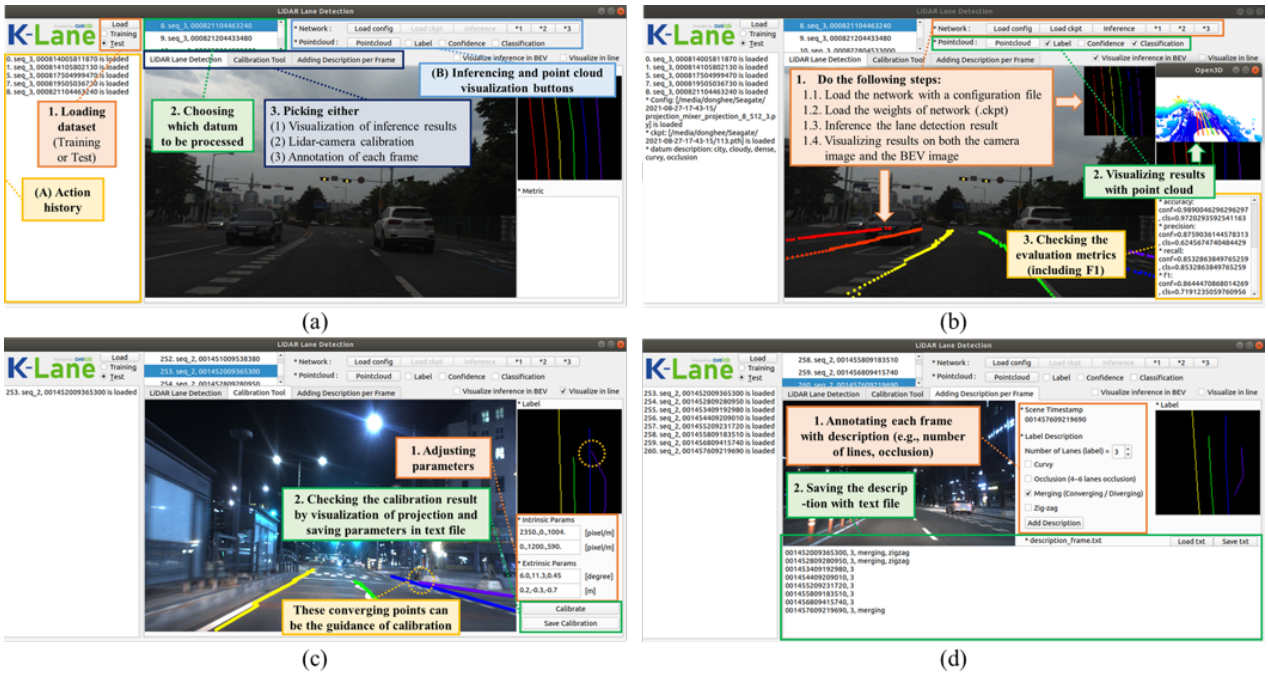


Figure 9. GUI-based Development Tools (GDT): (a) overall components of GDT and loading a data, (b) visualization of the LLDN inference results, (c) calibrating Lidar with camera (d) annotating a frame.

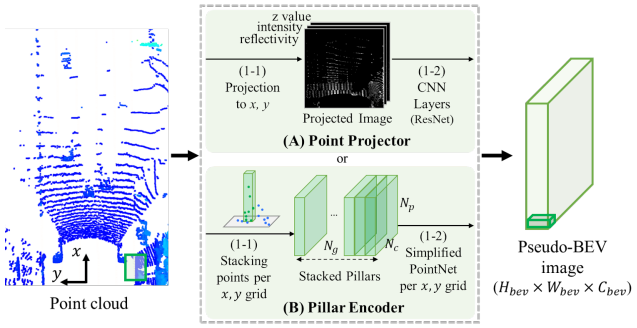


Figure 10. Detail structure of two BEV encoder: Point Projector and Pillar Encoder.

to generate three channels of the produced pseudo-BEV image. In order not to lose lane information while maintaining the real-time speed, we use only to a depth of the CNN where the feature map becomes the $1/8^2$ of the pseudo-BEV image input. To this end, we may use the first 14, 28, and 41 convolutional layers of the ResNet-18, ResNet-34, and ResNet-50 [5], respectively. Note that we denote these partial ResNets as ResNet14, ResNet28, and ResNet41 in the ablation studies in Section C, and that ResNet28 is the one used for the proposed LLDN-GFC.

An alternative for low computational 2D BEV encoder is the pillar encoder based on Point Pillars [9] that has relatively small network size. Pillar encoder has slightly lower performance but faster inference speed than the CNN-based point projector. Therefore, in this paper, pillar encoder is presented for real-time applications. As shown in Fig. 10, the pillar encoder aligns the point cloud in each grid of the horizontal plane to generate stacked pillars of size $N_g \times N_c \times N_p$, where N_g is the total number of grids, N_c is the point feature components, and N_p is the maximum number of points present on the grid. Then, a simplified version of PointNet [10] consisting of shared MLP's of size $N_c \times C$ is applied to each grid to extract pseudo-BEV image of size $H_{bev} \times W_{bev} \times C$. In this paper, considering that a lane in the real-world has a width of about 16cm and stretches long in the longitudinal direction of the road, the grid size in the pseudo-BEV is set to 32cm in the longitudinal direction and 16cm in the lateral direction.

Global Feature Correlator. Due to the advantage of patch-wise self attention networks (i.e., calculating the correlation in high resolution between distant grids within the feature map) for Lidar lane detection, we utilize two types of patch-wise self-attention network for global correlation, ViT [2] and MLP-Mixer [15] to propose two possible GFCs, GFC-T (the main proposal) and GFC-M (the computational alternative), respectively. In this section, we provide the structure of those GFCs in detail.

Fig. 11 shows the details of the two types of GFC, GFC-T and GFC-M. Both of the two GFCs employ (2-1), (2-2), (4-1), and (4-2) functions, while GFC-T employs (A) Transformer blocks and, a low computational alternative, GFC-

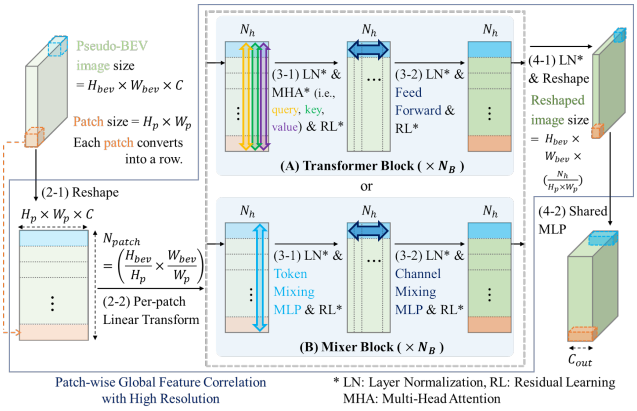


Figure 11. Details of proposed Global Feature Correlators; the input size is expressed with height H_{bev}, width W_{bev}, and number of channels C, and a patch size has height H_p and width W_p. The input and output size in Mixer block and Transformer encoder block is the same and the block repeats N_D times.

M uses (B) Mixer blocks for global correlation. In Fig. 11, function (2-1) reshapes the pseudo-BEV image into a 2D tensor for global correlation. Function (2-2) performs per-patch linear transform, and functions (3-1) and (3-2) perform global correlation through per-channel MLPs (i.e., Multi-head attention or Token-mixing MLP) and per-patch MLPs (i.e., Feed forward or Channel-mixing MLP), respectively. The Transformer encoder block in (A) performs global correlation between image patches using three MLPs calculating query, key, and value and utilizes the global correlation result to pay more attention (i.e., larger attention score) to the important patches to improve the global feature extraction. In addition, the Transformer encoder block allows visualization of the attention score, which can be used for analyzing the network inference, as shown in Appendix H. However, since the Transformer encoder block becomes a large network for the three MLPs and repeats calculating the attention score for every query (i.e., patch), the total computational cost increases in quadratic with the number of patches. On the other hand, Mixer block in (B) replace the multi-head attention, (3-1) in (A), with a single MLP, (3-1) in (B), which allows smaller network size and lower computational cost but it becomes difficult to analyze the network through attention score and causes lower model inductive bias than the Transformer block. Nonetheless, the two types of GFCs (GFC-T and GFC-M) show strong performance improvement in the Lidar lane detection.

Function (4-1) reshapes the last output of Transformer encoder and Mixer block to the size required for the lane detection head. Note that ViT and MLP-Mixer for the conventional image classification compress the detected feature with a classification token and global average pooling, respectively, but the proposed GFC reshapes the size up to $H_{bev} \times W_{bev}$ that is the input size to the function (2-1). This is how the proposed GFC provides inductive bias to the output feature map, which is testified with the visualized heatmap in Section 4.1, where high activation result is obtained in the resolution of pixels (much smaller resolution than patches). Note that the number of total pixels after the reshape becomes $H_p \times W_p$ times the number of total patches ($N_{patch} = H_{bev}/H_p \times W_{bev}/W_p$) before the reshape, which means that each pixel of the reshaped feature map has $N_h/(H_p \times W_p)$ dimension as a result. Since the channel size of the reshaped output image depends on the hidden dimension N_h, it can be smaller than that of the input BEV image, C_{bev}. This may cause bottleneck [14], so function (4-2) applies 1x1 convolution and produces the final output feature map for the detection head.

Detection Head. Fig. 12 shows the detection head introduced in Section 3.2. There are two segmentation heads: the classification head and the confidence head. Given an input of $H_{bev} \times W_{bev} \times C_{out}$ feature map from the GFC, we employ two sequential shared-MLPs to create the final prediction maps output. The first shared-MLP expands the dimension of the feature map from C_{out} to $2C_{out}$ for both classification and confidence heads to increase the representation capacity. The second shared-MLP then transforms the feature maps from $2C_{out}$ to N_{cls} and from $2C_{out}$ to 1 for the classification head and confidence head, respectively, resulting in a classification map and confidence map predictions. We then apply a grid-wise softmax to the classification map to get the $H_{bev} \times W_{bev} \times N_{cls}$ classification map output, and a grid-wise sigmoid to the confidence map to get the $H_{bev} \times W_{bev} \times 1$ confidence map output. The classification map shows per-class-probabilities of each grid, while the confidence map only shows the probability of the grid being a lane or not. The implementation of both classification and confidence tasks in parallel enables the LLDN to simultaneously predict the lane shape and the lane class.

As stated in Section 3.2, we use the soft dice loss [13]

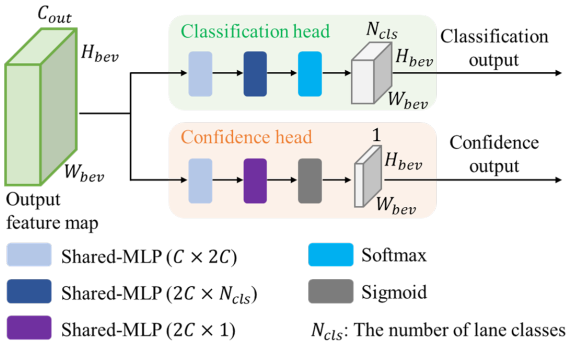


Figure 12. Detection head of the proposed LLDN-GFC.

for supervising the confidence loss $\mathcal{L}_{conf}$, defined as

$$\mathcal{L}_{conf} = 1 - \frac{2\sum_i^N \sum_j^M x_{conf_{i,j}} \hat{x}_{conf_{i,j}}}{\sum_i^N \sum_j^M x_{conf_{i,j}}^2 + \sum_i^N \sum_j^M \hat{x}_{conf_{i,j}}^2 + \epsilon}, \quad (1)$$

where ϵ is set to be 10^{-12} to prevent division by zero. The grid-wise cross-entropy loss [11] is used as the classification loss $\mathcal{L}_{cls}$, defined as

$$\mathcal{L}_{cls} = \frac{1}{NM} \sum_i^N \sum_j^M log(p(\hat{\boldsymbol{x}}_{cls_{i,j}})), \quad (2)$$

where $p(\hat{\boldsymbol{x}}_{cls_{i,j}})$ is the softmax of the classification prediction for class $k = x_{cls_{i,j}}$ at grid (i, j), defined as

$$p(\hat{\boldsymbol{x}}_{cls_{i,j}}) = \frac{exp(\hat{x}_{cls_{i,j,k}})}{\sum_{k'=1}^C exp(\hat{x}_{cls_{i,j,k'}})}. \quad (3)$$

The grid-wise cross-entropy loss penalizes the network based on the deviation of $p(\hat{\boldsymbol{x}}_{cls_{i,j}})$ from 1, which is equivalent to maximizing the probability of the correct class for each grid on the final classification map output. The total loss function $\mathcal{L}_{total}$ is the summation of both classification loss and the confidence loss as

$$\mathcal{L}_{total} = \mathcal{L}_{conf} + \mathcal{L}_{cls}. \quad (4)$$

B.2. Details of CNN-based LLDN

As we introduce in Section 4.1, we consider three types of CNN-based backbone as the counterparts of GFC, ResNet and FPN (RNF)-based backbone: (1) RNF-S, (2) RNF-D, and (3) RNF-C, where S, D, and C represent residual blocks implemented with strided convolution, dilated convolution, and convolutional block attention module (CBAM) [16], respectively. As shown in Fig. 6, there are 5 residual blocks composed of 3, 5, 5, 5, and 5 convolutional layers in the ResNet side. Each block produces a feature map that is 2^2 times smaller than the input feature map, and the feature pyramid network (FPN) concatenates the feature maps from each block to produce the final output feature map.

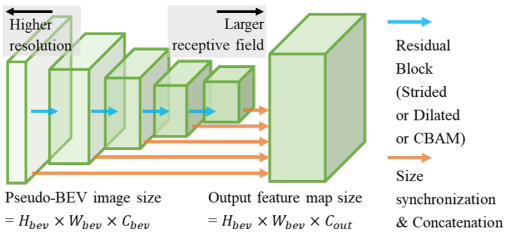


Figure 6. Overall structure of the conventional CNN-based backbone.

C. Ablations

In this section, we perform several ablation studies on the proposed LLDN-GFC, the low computational alternative, and the the conventional CNN-based LLDNs.

Ablations on Network Depth. Since hyperparameters related to the depth of LLDN are BEV encoder depth, the depth of backbone, we provide ablation studies in the following tables (Table 6∼9) to compare the performance of the LLDN with various BEV encoder, such as Proj14, Proj28, Proj41, and Pillars, and the depth of backbone. Tables in this subsection shows F1-score on the confidence (upper value) and that on the classification (lower value) in each table bin. FPS stands for frame per second, representing the overall computational cost of inference.

As shown in the Table 7, when we use Proj28 and increase the depth of GFC-T from GFC-T1 to GFC-T3, the performance increases by +2.3 and +2.3 in F1-scores on the confidence and the classification, respectively. In addition, as shown in the Table 6 and Table 7, when we use GFC-T3 and varies BEV depth from Proj14 to Proj28, the performance increases by +1.1 and +2.2 in F1-scores on the confidence and the classification, respectively. However, performance degradations are observed when the depth of GFC-T is increased from GFC-T3 to GFC-T5 for Proj28 and when depth of BEV encoder is increased from Proj28 to Proj41 for GFC-T3. From our ablation studies, we find that the model with an appropriate capacity can be Proj28-GFC-T3 for the proposed LLDN-GFC, Pillars-GFC-M5 for the low-computational alternative, and Proj28-RNF-S13, Proj28-RNF-C13, Proj28-RNF-D23 for the LLDNs using the conventional CNN-based backbones.

Note that for models with larger capacities, some regularization methods or more sophisticated learning techniques may be applied to reduce overfitting. However, those learning techniques are out of the scope of our study, since we focus on the network architecture and dataset.

Ablations on Hidden Dimension. As shown in the Table 10 and 11, we perform ablation studies on different hidden dimension size for Proj28-GFC-T3 and Pillars-GFC-M5, which are the best performing model of the proposed LLDN-GFC and its low computational alternative, respectively. As denoted in Section B.1, the hidden dimension N_h is the number of channels for each patch after the per-patch linear transform, indicating that higher value of hidden dimension leads to higher model capacity per each grid.

Table 10 shows the performance for various hidden dimension N_h of Proj28-GFC-T3; N_h=512 outperforms other variants, such as N_h=128 and N_h=2048. On the other hand, since Pillars-GFC-M5 requires more model capacity per each grid than Proj-GFC-T, Pillars-GFC-M with N_h=2048 outperforms that with N_h=512.

Ablations on Patch Size. We also perform ablation studies on the patch size of the Proj28-GFC-T3 and Pillars-GFC-

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gentle Curve	Sharp Curve	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
GFC	77.5	77.3	77.8	76.1	79.1	78.2	78.1	70.3	77.3	78.5	76.9	76.5	73.1	69.8	12.7
-T1	76.1	76.1	76.1	74.6	77.9	76.9	76.6	68.4	76.5	77.1	74.9	75.7	73.1	69.0	
GFC	**81.0**	**81.0**	**80.9**	**80.1**	**82.0**	**82.0**	**81.7**	**75.2**	**79.8**	**81.8**	**80.5**	**80.1**	**77.2**	**75.7**	12.6
-T3	**80.1**	**80.4**	**79.7**	**79.0**	**81.3**	**81.0**	**81.0**	**74.0**	**79.1**	**80.8**	**79.2**	**79.8**	**76.4**	**75.4**	
GFC	74.8	74.7	74.9	72.9	77.1	74.2	77.0	66.1	73.1	75.5	74.5	75.2	70.6	65.0	16.1
-M1	73.4	73.6	73.3	71.2	76.1	74.2	75.7	63.7	72.2	74.1	72.2	74.8	69.3	64.1	
GFC	78.9	79.0	78.9	77.5	80.6	79.5	80.4	72.2	77.6	79.8	78.3	78.5	74.7	70.1	15.4
-M3	77.8	78.0	77.6	76.2	79.8	78.4	79.4	70.5	76.7	78.7	76.8	77.8	73.8	69.4	
RNF	74.6	73.3	76.0	73.2	76.2	74.8	76.1	69.6	75.8	76.5	73.9	71.7	66.3	61.8	**16.5**
-S8	58.0	58.0	58.0	58.8	57.1	58.3	57.3	54.9	62.0	60.8	55.0	54.2	50.3	42.6	
RNF	77.7	76.4	79.3	76.3	79.4	78.0	79.6	72.2	78.0	79.5	77.0	75.0	70.7	66.9	15.5
-C8	63.7	62.5	65.0	62.6	65.0	63.9	64.7	58.1	65.9	66.3	60.9	59.6	56.7	48.6	
RNF	74.8	73.4	76.5	73.1	77.0	74.8	77.6	70.1	75.9	76.7	74.6	72.2	65.9	62.0	15.3
-D8	55.4	55.5	55.4	56.1	54.6	55.5	55.3	52.7	60.7	57.7	53.1	53.0	47.4	42.1	
RNF	67.4	65.9	69.2	66.4	68.7	67.3	69.6	63.9	69.6	69.2	67.0	64.3	59.6	56.8	15.0
-S13	62.0	60.9	63.3	61.4	62.7	61.8	64.1	59.1	65.6	63.9	61.0	59.0	55.2	50.2	
RNF	78.0	77.1	79.0	77.1	79.2	78.4	79.2	72.2	78.7	79.5	77.4	76.1	71.6	66.6	14.9
-C13	69.2	69.2	69.3	68.5	70.2	69.7	70.4	62.7	70.6	71.2	67.1	67.4	63.0	54.6	
RNF	76.9	75.7	78.2	75.5	78.5	77.0	79.0	71.5	77.8	78.7	76.2	74.3	69.1	63.6	14.8
-D13	60.4	60.2	60.7	61.7	58.9	60.8	59.7	56.21	65.5	62.9	57.6	57.1	53.8	46.7	

Table 6. Proj14-based LLDN performance for backbones with various depth.

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gentle Curve	Sharp Curve	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
GFC	79.8	79.6	80.0	79.4	80.3	80.2	80.6	75.2	79.4	80.8	79.1	78.7	74.9	72.8	11.8
-T1	78.8	78.9	78.7	78.3	79.4	79.3	75.2	73.5	78.4	79.8	77.7	78.2	73.9	72.5	
GFC	**82.1**	**82.2**	**82.0**	**81.7**	**82.5**	**82.5**	82.2	**78.0**	**81.0**	**82.9**	**81.4**	**82.3**	**78.7**	75.9	11.6
-T3	**81.1**	**81.4**	**80.7**	**80.6**	**81.7**	**81.5**	**83.0**	**76.7**	**80.1**	**81.9**	**81.4**	**81.3**	**78.7**	75.5	
GFC	81.1	81.0	81.2	80.6	81.7	82.0	**82.3**	76.0	80.0	82.1	80.5	79.6	77.3	**77.2**	11.2
-T5	79.5	79.5	79.4	78.7	80.4	80.0	80.7	73.0	78.8	80.3	78.5	78.6	75.3	**76.2**	
GFC	78.5	78.5	78.4	77.8	79.3	78.9	80.0	72.5	78.0	79.4	77.8	77.7	74.5	70.2	**13.4**
-M1	77.3	77.6	77.0	76.4	78.4	77.8	79.1	70.2	76.9	78.2	77.8	77.7	74.5	69.5	
GFC	79.7	79.9	79.6	78.9	80.8	80.1	81.3	74.6	79.0	80.4	79.6	79.4	76.1	72.5	13.3
-M3	78.8	79.0	78.4	77.7	80.0	79.2	80.4	72.6	78.1	79.4	78.3	78.9	74.9	71.7	
GFC	78.7	77.3	78.8	78.0	79.6	79.0	80.5	73.5	77.6	79.6	78.2	78.1	74.7	69.9	13.1
-M5	79.2	79.5	78.9	78.4	80.1	79.0	81.1	73.6	78.4	80.1	78.6	78.6	75.3	71.4	
RNF	74.6	73.9	75.4	74.4	74.9	74.9	75.3	70.5	76.0	76.5	73.5	71.8	66.9	64.8	13.2
-S8	63.0	62.8	63.3	63.4	62.6	63.5	62.7	57.6	66.7	65.3	60.3	60.9	54.7	49.8	
RNF	78.1	77.3	79.1	77.6	78.7	78.2	79.7	74.9	79.0	79.7	77.3	76.1	71.5	68.6	13.1
-C8	70.3	69.7	71.0	69.8	70.9	70.4	71.6	66.6	72.0	72.2	68.0	69.0	62.8	58.5	
RNF	73.2	72.6	74.0	73.1	73.3	73.3	74.0	70.5	74.8	74.9	72.2	70.9	65.7	63.5	13.1
-S13	70.5	70.1	71.0	70.4	70.6	70.4	71.9	68.1	72.5	72.3	68.4	69.0	63.3	59.0	
RNF	78.0	77.6	78.5	77.7	78.3	77.9	80.0	76.0	78.9	79.6	76.9	76.0	71.9	69.3	13.0
-C13	75.3	75.1	75.5	74.8	76.0	75.0	77.9	73.1	76.5	77.0	73.1	74.1	69.2	65.3	
RNF	69.5	68.4	70.8	69.5	69.6	69.6	70.1	67.3	72.1	71.6	68.5	66.2	61.5	58.5	13.1
-D13	65.5	64.9	66.2	65.6	65.3	65.5	65.8	62.9	68.7	67.6	63.1	63.0	58.5	54.6	
RNF	72.1	71.3	73.0	71.9	72.3	72.2	72.9.	69.6	74.0	74.0	70.9	69.5	63.8.	61.9	12.7
-D23	68.8	68.3	69.4	68.7	69.0	68.8	69.8	66.5	71.8	70.7	66.8	67.2	61.1	57.6	

Table 7. Proj28-based LLDN performance for backbones with various depth.

M5. The results in Table 10 and 11 show that there is a significant performance drop as P is increased from 8 to 16. This is because when the patch size is increased to 16 (from 8), the number of grids covered by a patch increases four

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gentle Curve	Sharp Curve	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
GFC	77.8	77.7	77.9	77.5	78.2	78.3	78.5	72.3	78.0	78.9	76.7	76.5	74.2	70.0	11.0
-T1	76.4	76.7	76.1	76.2	76.6	77.0	76.9	70.4	77.2	77.4	75.0	75.6	73.0	69.7	
GFC	**80.5**	**80.6**	**80.5**	**80.4**	80.7	**81.2**	80.9	**75.2**	79.3	**81.4**	**79.8**	**79.4**	**77.3**	**75.0**	10.9
-T3	**79.1**	**79.4**	**78.8**	**79.1**	79.1	**79.8**	79.4	**73.4**	78.3	**79.9**	78.2	78.5	**75.9**	**74.5**	
GFC	78.2	78.3	78.1	77.0	79.7	78.8	79.6	71.5	77.2	79.0	77.6	78.1	73.9	71.2	**11.7**
-M1	77.1	77.4	76.7	75.7	78.9	77.8	78.4	69.4	76.3	77.9	77.5	77.5	72.7	70.6	
GFC	79.9	80.1	79.7	79.2	**80.9**	80.5	80.6	74.9	79.0	80.9	79.0	79.5	76.1	72.2	11.6
-M3	78.8	79.2	78.4	77.8	**80.1**	79.4	72.5	72.5	77.9	79.7	77.4	79.0	75.0	71.2	
GFC	79.7	79.8	79.7	79.0	80.7	80.1	**81.1**	74.6	**79.6**	80.6	79.4	79.1	75.7	70.6	11.4
-M5	78.8	79.0	78.6	77.9	79.9	79.2	**80.3**	72.7	**78.6**	79.6	**78.3**	**78.7**	74.8	69.9	
RNF	75.1	73.8	76.5	74.4	75.8	75.0	76.3	72.3	77.1	77.0	74.6	71.8	66.8	63.6	11.1
-S8	61.8	61.7	62.0	62.8	60.7	61.9	61.3	59.5	67.0	64.0	59.5	59.4	54.2	48.6	
RNF	76.0	75.0	77.1	75.1	77.1	76.0	77.5	72.4	77.8	77.8	74.9	73.7	68.1	65.9	10.7
-C8	69.4	68.7	70.1	68.1	70.9	69.3	71.2	65.7	71.7	71.4	66.8	67.5	62.5	56.9	
RNF	72.2	70.6	74.0	71.4	73.1	72.0	73.8	70.3	74.5	74.3	71.3	68.7	63.3	58.9	10.9
-D8	65.9	64.7	67.2	65.3	66.6	65.5	67.7	63.8	69.3	68.1	63.8	63.1	57.6	51.5	
RNF	69.8	68.8	70.9	69.7	69.9	70.0	70.7	67.1	72.4	71.7	69.1	66.6	61.7	57.6	10.8
-S13	66.3	65.5	67.1	66.3	66.2	65.1	66.5	63.8	69.6	68.5	64.2	63.8	58.5	51.4	
RNF	77.6	76.8	78.5	76.7	78.6	78.0	79.3	73.9	78.1	79.2	76.8	75.5	70.5	67.6	10.5
-C13	73.9	73.4	74.5	73.0	75.0	74.0	75.9	70.2	75.1	75.8	71.7	72.3	66.9	61.9.	
RNF	69.9	68.7	71.3	69.5	70.4	69.8	70.9	67.6	72.7	71.8	69.1	66.6	62.7	59.2	10.8
-D13	65.4	64.5	66.3	65.3	65.5	79.2	80.3	63.4	69.4	67.3	63.5	63.2	58.8	52.8	
RNF	69.9	68.7	71.4	69.6	70.3	70.0	70.7	67.2	72.1	71.9	69.1	66.7	62.0	58.7	10.3
-S23	66.4	65.6	67.3	66.0	66.9	66.0	67.8	63.0	69.1	68.5	64.3	63.6	60.0	54.6	
RNF	70.1	69.7	72.7	70.4	70.6	70.1	71.2	67.1	72.9	72.3	70.4	66.9	62.5	59.3	10.2
-D23	66.1	65.1	67.2	65.4	67.3	66.1	67.3	63.4	70.4	68.9	64.2	63.3	59.2	52.6	

Table 8. Proj41-based LLDN performance for backbones with various depth.

times, so that the GFC has to extract global features from a map with four times lower resolution.

D. Qualitative Results Visualization

In addition to the numerical results in Section 4.1, we provide qualitative results of the proposed LLDN-GFC, Proj28-GFC-T3, its low computational alternative, Pillars-GFC-M5, and the conventional CNN-based LLDNs, such as Proj28-RNF-S13, Proj28-RNF-C13, and Pillars-RNF-C13, using visualization.

D.1. Qualitative Results

Fig. 13∼16 in this subsection has 4 rows and 5 columns, where each row shows inference results for different scenes (conditions) and each column shows inference results for different GFCs. Each inference result has upper and lower plots for the projection of inference results into the front view image with true labels in the upper left corner and for the inference on top of the BEV point cloud, respectively.

Fig. 13 and 14 show inference results of LLDNs with Proj28 for scenes with moderate (e.g., normal, no occlusion, and gentle curve) and severe (e.g., occlusion, merging, and sharp curve) lane detection difficulties, respectively, while Fig. 15 and 16 show inference results of LLDNs with Pillars.

In all figures shown in this subsection, LLDNs based on GFC-T and GFC-M (shown in (a) and (b) of figures, respectively) show better performance than other LLDNs (in (c), (d), and (e) of figures) regardless of the lane detection difficulties and BEV encoders (i.e., Proj28 and Pillars). For example, plots in (a) and (b) show a strong lane detection performance even for images of severe occlusion, where a good portion of point cloud data are missing.

D.2. Qualitative Heatmaps

We emphasize the performance of the proposed GFC, GFC-T, and its low computational alternative, GFC-M, using the visualization of the heatmaps for occlusion scenes as shown in Fig. 6 in Section 4.1. In addition to the results in Fig. 6, we provide more visualization of heatmaps for various difficult scenes, such as curved lanes, merging lanes, and other severe occlusion cases, to emphasize the superior performance of the proposed GFC and its low-computational alternative GFC.

All of the figures in this subsection follow the same format used for Fig. 6 in Section 4.1. The four columns are

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gentle Curve	Sharp Curve	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
GFC	64.3	63.7	64.9	61.0	68.2	65.3	66.2	51.4	64.6	64.9	63.2	65.6	59.8	56.1	14.0
-T1	62.4	61.9	62.9	58.9	68.2	63.5	64.5	48.5	63.3	63.1	60.3	64.4	58.1	54.4	
GFC	76.2	76.2	76.1	73.6	**79.2**	76.8	78.7	67.0	75.0	76.7	75.5	77.1	72.2	69.2	13.9
-T3	74.7	75.0	74.4	72.0	78.1	75.4	77.3	64.8	74.0	75.3	73.1	76.5	71.0	68.6	
GFC	**78.5**	**78.5**	**78.4**	**77.8**	**79.2**	**77.9**	**79.0**	**72.5**	**78.0**	**79.4**	**77.8**	**77.7**	**74.5**	**70.2**	13.8
-T5	**77.3**	**77.6**	**77.0**	**76.4**	**78.4**	**77.6**	**77.8**	**70.2**	**76.9**	**78.2**	**76.3**	**77.2**	**73.2**	**69.5**	
GFC	64.5	63.7	65.5	59.1	71.0	65.3	69.8	50.6	59.9	65.1	63.9	66.7	60.0	50.4	**16.6**
-M1	62.9	62.1	63.8	57.2	69.7	63.8	68.4	47.7	59.0	63.6	61.3	65.6	58.4	48.7	
GFC	70.0	69.8	70.1	66.0	74.7	71.0	73.7	56.7	66.4	70.6	69.4	72.0	64.6	57.5	16.4
-M3	60.6	61.2	59.8	56.8	65.1	62.0	63.6	44.3	57.8	61.6	58.1	63.0	56.3	42.9	
GFC	74.8	74.8	74.9	72.0	78.2	75.6	77.9	64.6	72.0	75.5	74.4	76.0	69.3	65.2	16.3
-M5	73.5	73.6	73.4	70.5	77.1	74.4	76.6	62.2	71.1	74.2	72.3	75.5	67.6	62.3	
RNF	70.1	69.3	71.1	67.6	73.2	69.9	75.7	62.9	70.2	71.5	70.9	69.0	61.9	50.9	15.9
-C8	22.1	22.3	21.9	25.1	18.5	23.2	19.1	17.4	23.1	25.8	16.9	16.6	12.7	8.5	
RNF	64.6	62.9	66.5	59.4	70.7	51.1	72.1	51.2	63.4	65.5	65.9	65.0	56.9	44.9	15.7
-S13	18.2	16.4	20.4	15.6	21.4	13.1	22.6	13.1	16.0	19.2	18.7	16.8	16.0	4.7	
RNF	76.8	75.9	77.8	74.5	79.6	67.6	81.4	67.6	76.6	77.9	77.9	75.7	69.5	62.5	15.5
-C13	40.6	39.1	42.4	40.6	40.6	32.5	43.6	32.5	42.6	43.6	38.3	35.7	32.8	20.4	
RNF	61.4	60.0	62.9	56.8	66.8	61.0	69.5	51.8	60.0	62.3	62.7	61.2	53.8	42.6	15.4
-D13	16.5	14.8	18.4	13.5	20.0	15.8	22.6	12.2	13.9	17.4	16.7	14.5	14.2	4.5	
RNF	58.5	56.9	60.4	52.6	65.6	59.2	64.9	42.2	55.8	59.1	59.1	59.9	51.5	43.6	15.3
-S23	31.9	29.9	34.2	28.4	36.1	32.0	42.2	23.2	30.2	32.6	32.1	32.0	28.5	17.3	
RNF	63.2	61.6	65.0	58.6	68.6	51.1	70.7	51.3	62.2	64.2	64.3	63.1	55.1	43.9	15.2
-D23	19.0	17.0	21.2	17.2	21.1	13.5	23.7	13.5	17.2	20.2	20.2	19.1	16.7	4.9	

Table 9. Pillars-based LLDN performance for backbones with various depth.

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gen. Cur.	Sha. Cur.	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
$P8$	**82.1**	**82.2**	**82.0**	**81.7**	**82.5**	**82.5**	82.2	**78.0**	**81.0**	**82.9**	**81.4**	**82.3**	**78.7**	**75.9**	11.6
N_h512	**81.1**	**81.4**	**80.7**	**80.6**	**81.7**	**81.5**	**83.0**	**76.7**	**80.1**	**81.9**	**81.4**	**81.3**	**78.7**	**75.5**	
$P16$	80.2	79.3	81.2	78.4	82.4	81.0	82.4	73.3	78.9	80.7	80.5	79.7	76.6	75.6	**11.9**
N_h512	78.1	77.6	78.7	75.8	80.9	79.0	80.5	70.4	76.9	78.5	77.7	78.5	75.1	74.6	
$P8$	81.5	81.5	81.5	81.1	82.0	81.9	**82.8**	76.9	80.2	82.5	81.2	80.6	76.0	74.3	11.8
N_h128	75.3	75.7	74.9	74.5	76.4	76.2	76.6	66.2	74.6	76.3	73.9	75.5	70.3	68	
$P8$	76.6	75.9	77.4	75.5	77.8	77.4	78.3	67.3	76.1	77.8	76.7	74.9	71.3	64.7	10.7
N_h2048	61.2	60.4	62.1	59.6	63.1	62.3	61.9	50.2	62.6	63.1	58.5	60.1	55.5	44.9	

Table 10. Performance of Proj28-GFC-T3 for various hidden dimension and patch sizes, where P and N_h represent the patch size and the hidden dimension size with default value 8 and 512, respectively.

Back-bone	Total	Day	Night	Ur-ban	High-way	Nor-mal	Gen. Cur.	Sha. Cur.	Mer-ging	No Occ	Occ 1	Occ 2	Occ 3	Occ 4-6	FPS
$P8$	74.8	74.8	74.9	72.0	78.2	77.9	75.6	**64.6**	72.0	75.5	74.4	76.0	69.3	65.2	16.3
N_h512	73.5	73.6	73.4	70.5	77.1	74.4	76.6	62.2	71.1	74.2	72.3	75.5	67.6	62.3	
$P16$	72.2	72.0	72.4	68..8	76.2	67.0	75.9	58.4	70.2	72.6	71.5	73.4	68.9	63.4	**16.6**
N_h512	65.2	65.2	65.2	61.8	69.3	73.0	67.2	49.5	64	66.2	62.8	66.8	60.5	55.8	
$P8$	70.9	70.9	71.0	67.7	74.8	71.7	74.6	59.4	68.4	71.6	70.2	72.5	65.7	60.2	16.5
N_h128	63.1	62.7	63.4	59.4	67.4	64.1	66.0	49.2	61.7	64.1	60.7	65.4	56.7	48.5	
$P8$	**75.6**	**75.5**	**75.6**	**72.9**	**78.8**	**76.4**	**78.8**	64.5	**73.5**	**76.2**	**74.8**	**76.6**	**71.2**	**66.5**	14.7
N_h2048	**74.1**	**74.4**	**73.8**	**71.2**	**77.6**	**75.0**	**77.2**	**62.6**	**72.6**	**74.8**	**72.5**	**76.1**	**69.7**	**68.2**	

Table 11. Performance of Pillars-GFC-M5 for various hidden dimension and patch sizes; where P and N_h represent the patch size and the hidden dimension size with default value 8 and 512, respectively.

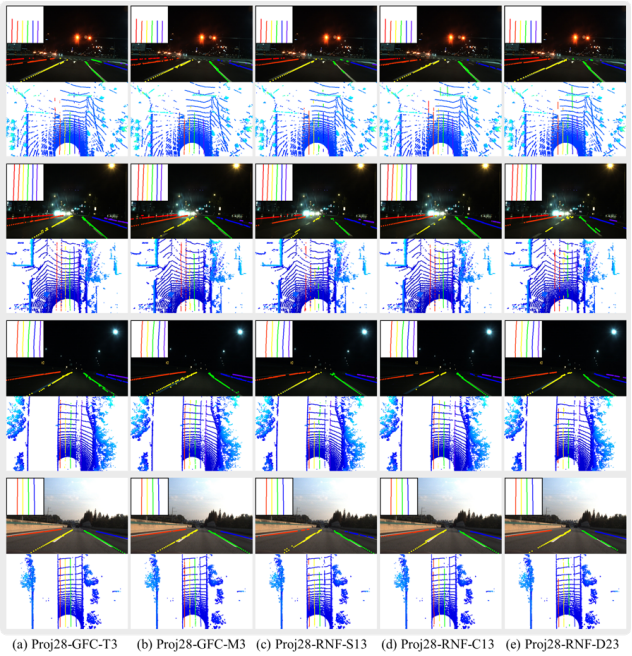


Figure 13. Lane detection performance comparison for LLDNs with Proj28 for images with moderate difficulty (e.g., normal, no occlusion, and gentle curve).

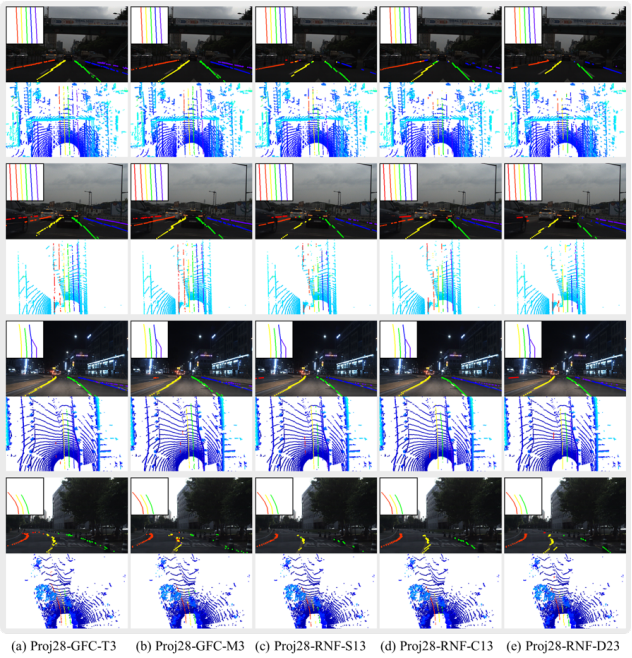


Figure 14. Lane detection performance comparison for LLDNs with Proj28 for images with high difficulty (e.g., occlusion, merging, and sharp curve).

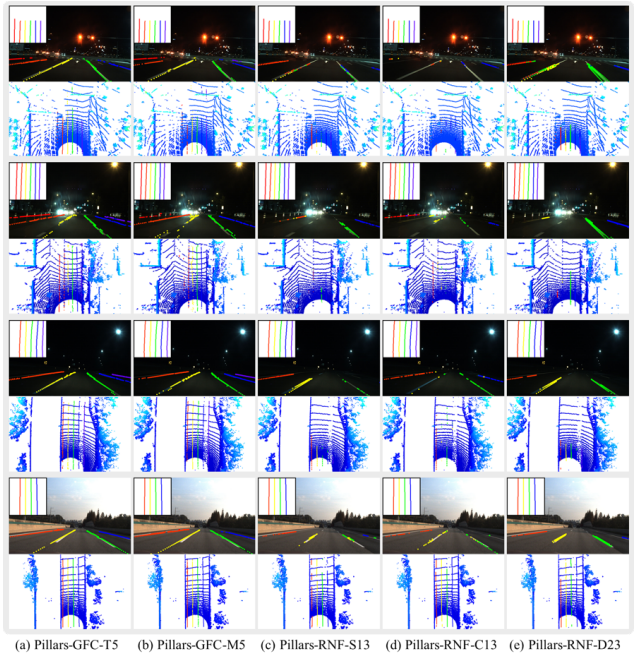


Figure 15. Lane detection performance comparison for LLDNs with Pillars for images with moderate difficulty (e.g., normal, no occlusion, and gentle curve).

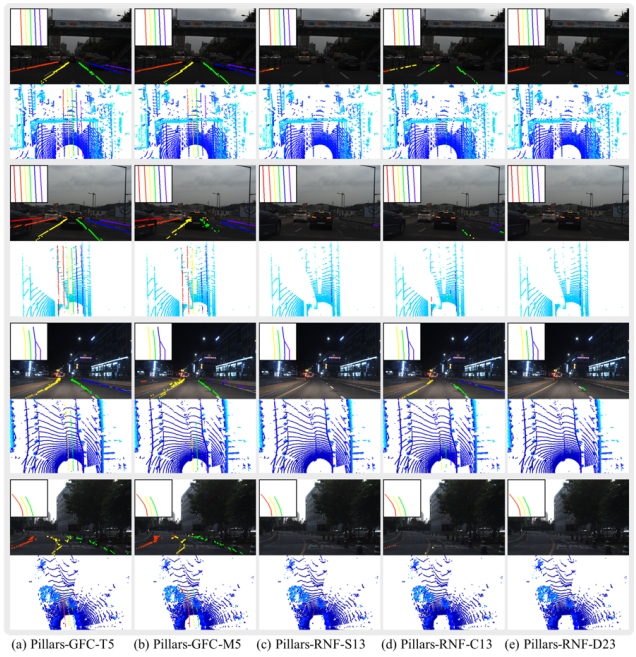


Figure 16. Lane detection performance comparison for LLDNs with Pillars for images with high difficulty (e.g., occlusion, merging, and sharp curve).

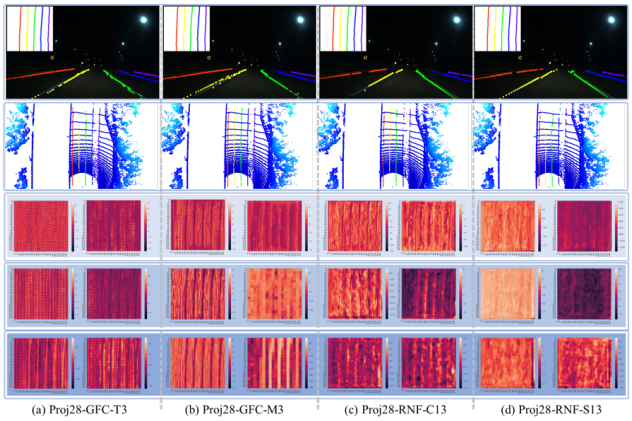


Figure 17. Comparison of the lane detection performance of the proposed LLDN-GFC, Proj28-GFC-T3, to the Proj28-GFC-T3 and other CNN-based LLDNs (Proj28-RNF-C13 and Proj28-RNF-S13) for curved lanes.

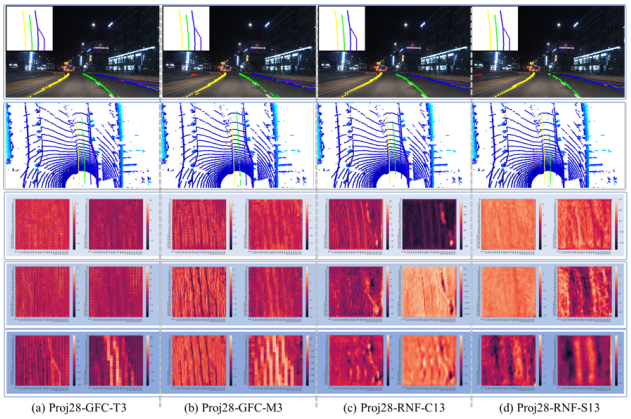


Figure 18. Comparison of the lane detection performance of the proposed LLDN-GFC, Proj28-GFC-T3, to the Proj28-GFC-M3 and other CNN-based LLDNs (Proj28-RNF-C13 and Proj28-RNF-S13) for merging lanes.

inference results for (a) Proj28-GFC-T3, (b) Proj28-GFC-M3, (c) Proj28-RNF-C13, and (d) Proj28-RNF-S13. The first row shows the projection of inference results into the front view image with true labels in the upper left corner, and the second row shows the inference on top of the BEV point cloud. From the 3rd to 5th row, we show the heatmap of the 1st, 2nd, and 3rd block output feature map of the GFC (e.g., 1st, 2nd, and 3rd Transformer block of Proj28-GFC-T3). Output feature maps at different blocks are resized or reshaped (i.e., in the same way to the function (4-1) in Fig. 11) and two heatmaps are sampled along the channels.

As shown in Fig. 17, Fig. 18, and Fig. 19, both the proposed LLDN-GFC, Proj28-GFC-T3, and the LLDN with the low computational alternative GFC, GFC-M3, demonstrate three advantages described in Section 4.1 for curved

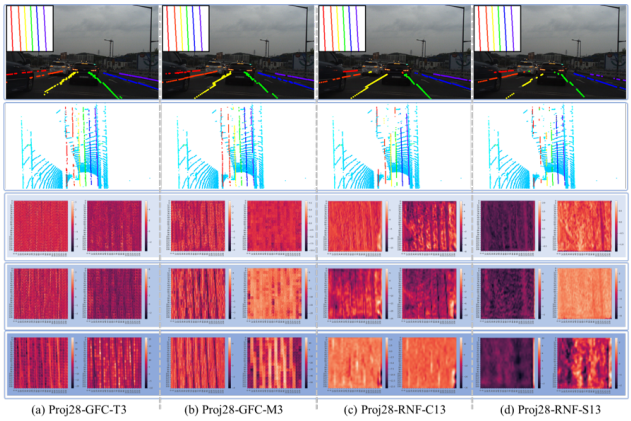


Figure 19. Comparison of the lane detection performance of the proposed LLDN-GFC, Proj28-GFC-T3, to the Proj28-GFC-M3 and other CNN-based LLDNs (Proj28-RNF-C13 and Proj28-RNF-S13) for occluded lanes.

lanes, merging lanes, and other occluded lanes. The three advantages are (1) better resolution as the network deepens, (2) distinctive color difference between lane and non-lane positions, and (3) predicting the shape of the lane even in presence of occlusion.

E. LLDN vs Heuristic Method

In the heuristic Lidar lane detection techniques, we first project pointcloud into a BEV image and apply thresholding to remove low-intensity points [6]. The remaining points are then clustered using, for example, DBSCAN [3] and then fitted by the first order polynomial to create smooth lane lines.

In the experiments, we observe multiple instances when the heuristic technique is unreliable; First, when a strong light illuminates a spot on the ground, as shown in Fig. 20 (b), it results in false positives (FPs). Second, when lane marks are occluded, the heuristic Lidar lane detection can-

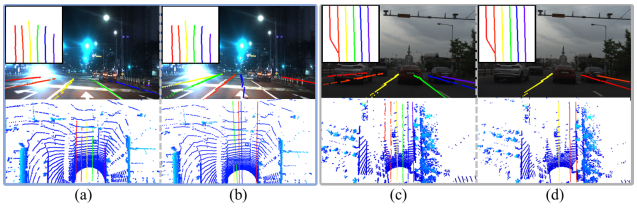


Figure 20. Comparison between the proposed LLDN-GFC (Proj28-GFC-T3) (in (a) and (c)) and heuristic Lidar lane detection (in (b) and (d)). When a strong source of illumination appears on the scene, (b) the heuristic method fails, but (a) the proposed LLDN-GFC is not affected. When lane marks are occluded, (d) the heuristic method cannot infer the lanes, but (c) the proposed LLDN-GFC is able to infer the occluded lane lines.

not infer the presence of lane marks, leading to a high false negatives (FNs), as shown in Fig. 20 (d). However, the proposed LLDN-GFC can produce reliable lane detection results for the two scenarios. As the LLDN-GFC learns global context features of the scene, a bright illuminated road spot or partial occlusion of lane lines hardly degrade the lane detection performance.

References

[1] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020. 2

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 3

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 10

[4] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[6] Danilo Caceres Hernandez, Van-Dung Hoang, and Kang-Hyun Jo. Lane surface identification based on reflectance using laser range finder. In *2014 IEEE/SICE International Symposium on System Integration*, pages 621–625. IEEE, 2014. 10

[7] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011. 2

[8] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[9] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 3

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[12] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[13] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. *Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations*. Springer, 2017. 4

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4

[15] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021. 3

[16] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5