GLane3D : Detecting Lanes with Graph of 3D Keypoints

Supplementary Material

```python
def PointNMS(points_x, points_y, scores,
             thresh_x, thresh_y, r=10):
    x1 = points_x * r - (r / 2) * thresh_x
    x2 = points_x * r + (r / 2) * thresh_x
    y1 = points_y * r - (r / 2) * thresh_y
    y1 = points_y * r + (r / 2) * thresh_y
    boxes = stack(round(x1), round(y1),
                  round(x2), round(y2))

    keep = BoxNMS(boxes, scores, iou_thresh=0.1)
    return keep
```

Figure 6. Python code of PointNMS function

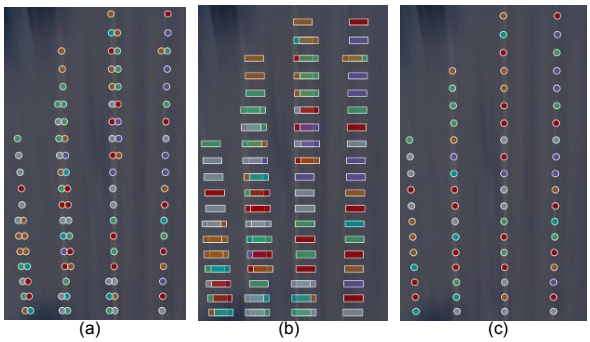

Figure 7. PointNMS Process: (a) Initial proposals, (b) Bounding boxes centered at the proposal points, (c) Selected strongest keypoints after applying PointNMS.

6. Hyperparameter Relationship: N and S in Proposal Selection

The number of proposals, N, serves as a hyperparameter in GLane3D, determining the total proposals selected from the set of anchor keypoints $\mathbf{K}_A$. Our model leverages multiple proposals per keypoint to effectively represent the target lane. These proposals are refined using PointNMS, a function that retains the strongest S non-overlapping keypoints. Since each keypoint is chosen from n proposals, where n represents the number of proposals per target keypoint, the relationship $N = S \times n$ is adhered to during the hyperparameter selection process.

7. Cross Dataset Evaluation

Tab. 9 provides an extended version of Tab. 7, presenting a more comprehensive comparison. The results demonstrate that GLane3D achieves superior performance in cross-dataset evaluations.

The gap in the 0.5m thresholded F1-score highlights the superior generalization capability of our model. Specifically, while GLane3D achieves a +5.7 improvement in F1-score under the 1.5m threshold, the notable +25.2 improvement in the 0.5m threshold further underscores its enhanced generalization compared to previous methods.

The qualitative results in Fig. 9 help to explain the significant F1-score gap under the 0.5m matching threshold. These results reveal that the localization performance of previous methods is inferior to that of GLane3D, particularly in precise lane representation.

8. PointNMS Implementation

The PointNMS function selects only the highest-confidence keypoint from a group of proposals located within a distance of d_x from each other, with d_y being approximately zero.

Rather than implementing a custom module for the PointNMS operation, we leverage the commonly available BoxNMS function to achieve the same result.

The input to the function consists of the proposal locations ($\mathbf{K}_P$), as depicted in Fig. 9a. Bounding boxes are

Dist.	Methods	Backbone	F1-Score↑	AP(%) ↑	X-error near(m) ↓	X-error far(m) ↓	Z-error near(m) ↓	Z-error far(m) ↓
1.5m	PersFormer [2]	EffNet-B7	53.2	-	0.407	0.813	0.122	**0.453**
	LATR [22]	ResNet-50	34.3	51.2	0.327	0.737	0.142	0.501
	GLane3D-Base (Ours)	ResNet-50	54.9	**64.0**	**0.255**	**0.694**	**0.078**	0.464
	GLane3D-Large (Ours)	Swin-B	**58.9**	64.9	0.289	0.701	0.086	0.479
0.5m	PersFormer [2]	EffNet-B7	17.4	-	0.246	0.381	0.098	0.214
	LATR [22]	ResNet-50	19.0	27.8	0.201	0.313	0.116	0.220
	GLane3D-Base (Ours)	ResNet-50	40.7	44.2	**0.135**	0.297	**0.055**	**0.194**
	GLane3D-Large (Ours)	Swin-B	**42.6**	**44.4**	0.162	**0.296**	0.063	0.198

Table 9. Extended Cross-Dataset Evaluation on the Balanced Scenes of the Apollo Dataset [5].

then generated around these proposals, as shown in Fig. 9b, where the centers of the bounding boxes correspond to the positions of the proposals. The width of each box is set to d_x, and the height is d_y. To enhance resolution, the proposal positions are scaled using a reshaping parameter r, as illustrated in Fig. 6.

9. Spatial Constraints in the Matcher

We use the Hungarian algorithm to match predicted proposals with ground truth keypoints, as discussed in Sec. 3.5. GLane3D utilizes anchor points $\mathbf{K}_A$ during the prediction of proposals $\mathbf{K}_P$, where the anchor points maintain a fixed initial position that is not updated. Because of this, we apply spatial constraints to avoid matching proposals with distant ground truth keypoints.

The first constraint concerns the lateral distance between the predicted position of a proposal, $(\mathbf{x}_i + \Delta x, \mathbf{y}_i)$, and the ground truth keypoint, $(\mathbf{x}_t, \mathbf{y}_t)$. If the lateral distance, $|\mathbf{x}_i + \Delta x - \mathbf{x}_t|$, exceeds 1 meter, the match is rejected. The second constraint addresses the initial position of the anchor point, $(\mathbf{x}_i, \mathbf{y}_i)$, and the ground truth position, $(\mathbf{x}_t, \mathbf{y}_t)$. If the lateral distance, $|\mathbf{x}_i - \mathbf{x}_t|$, exceeds 2 meters, the match is also rejected. The third constraint relates to the longitudinal distance between the proposal and the ground truth. Since our model does not predict an offset in the longitudinal y-axis, we reject matches if the longitudinal distance $|\mathbf{y}_i - \mathbf{y}_t|$ exceeds 0 meters.

We prevent matching when any of these conditions are met by replacing the corresponding value in the cost matrix with an infinite value.

10. Graph of Keypoints to Lane Instances

The output of GLane3D is a graph representing keypoints, where directed edges connect sequential keypoints, as shown in Fig. 8a. Initially, we identify the start and end keypoints, which satisfy the conditions specified in Eq. (7) and Eq. (8), respectively. These keypoints are represented as green and red points in Fig. 8a. In the next step, we compute the shortest paths between the start and end keypoints using Dijkstra's algorithm. To incorporate adjacency probabilities $\mathbf{A}$ into the shortest path estimation, we use the corresponding values from the matrix $1 - \mathbf{A}$, as this is a minimization problem.

11. Custom BEV Geometry Adjustment

As discussed in Sec. 3.2, the anchor points $\mathbf{K_A}$ used in Inverse Projection Mapping (IPM) are evenly distributed across the Bird's Eye View (BEV) space. However, when projected onto the frontal view (FV), the anchor points become sparser in regions closer to the ego vehicle and denser in areas farther away, as shown in Fig. 3a.

To address this sparsity near the ego vehicle, our method adjusts the distribution of anchor points $\mathbf{K_A}$ by reducing both the longitudinal and lateral distances between keypoints as they approach the ego vehicle. Specifically, the longitudinal distances between keypoints increase linearly from 0.5 to 1.5 meters, from the ego vehicle to the farthest point, as described in Eq. (10) and Eq. (9). Meanwhile, the lateral distances between keypoints at the same longitudinal distance decrease as they approach the ego vehicle. This adjustment narrows the width of the BEV space near the ego vehicle, ensuring that the number of columns remains constant while preserving the rectangular shape of the BEV feature F_{BEV}. As shown in Eq. (11) and Eq. (12), the points in the first row begin at $W \times \frac{1}{4}$ and end at $W \times \frac{3}{4}$, while points in the farthest row start at 0 and end at W. The lateral range increases linearly from the nearest row to the farthest row.

$$\text{dy} = \frac{1}{H-1} \tag{9}$$

$$y_i = 0.5 + i \times \text{dy}, \quad \text{for } i = 0, 1, \ldots, H-1 \tag{10}$$

$$x_{\text{start},i} = \frac{W}{4}\left(1 - \frac{i}{H-1}\right) \tag{11}$$

$$x_{\text{end},i} = W \times \frac{3}{4}\left(1 - \frac{i}{H-1}\right) + W \times \frac{i}{H-1} \tag{12}$$

12. PV to BEV: IPM vs LSS

We compare two commonly used projection methods—Inverse Projection Mapping (IPM) and Lift Splat Shoot (LSS)—with GLane3D. IPM projects Bird's Eye View (BEV) locations to the frontal view using camera parameters. After projection, features extracted from corresponding locations in the frontal view are sampled, resulting in $\mathbf{F_{BEV}} \in \mathbb{R}^{c \times H_b \times W_b}$.

Projection	F1-Score(%) ↑	X-error near(m) ↓	X-error far(m) ↓	Z-error near(m) ↓	Z-error far(m) ↓
IPM w/o Custom BEV	71.6	0.222	0.275	0.094	0.125
IPM w/ Custom BEV	72.0	0.239	0.267	0.093	0.121
LSS	72.1	0.219	0.261	0.091	0.118

Table 10. Comparison of PV to BEV Projection Methods.

Dist.	Methods	Backbone	Sensors	All	Up & Down	Curve	Extreme Weather	Night	Inter.	Merge Split
1.5m	PersFormer [2]	EffNet-B7	C	50.5	42.4	55.6	48.6	46.6	40.0	50.7
	BEV-LaneDet [42]	ResNet-34	C	58.4	48.7	63.1	53.4	53.4	50.3	53.7
	PersFormer [2]	ResNet-50	C	53.7	46.4	57.9	52.9	47.2	41.6	51.4
	M^2-3DLaneNet [21]	EffNet-B7	C+L	55.5	53.4	60.7	56.2	51.6	43.8	51.4
	LATR [22]	ResNet-50	C	61.9	55.2	68.2	57.1	55.4	52.3	61.5
	LaneCPP [28]	EffNet-B7	C	60.3	53.6	64.4	56.7	54.9	52.0	58.7
	PVALane [51]	ResNet-50	C	62.7	54.1	67.3	62.0	57.2	53.4	60.0
	DV-3DLane [23]	ResNet-34	C+L	65.4	60.9	72.1	64.5	61.3	55.5	61.6
	DV-3DLane [23]	ResNet-50	C+L	**66.8**	61.1	71.5	**64.9**	**63.2**	**58.6**	62.8
	Glane3D-Lite	ResNet-18	C	61.5	55.6	69.1	56.6	56.6	52.9	61.3
	GLane3D-Base	ResNet-50	C	63.9	58.2	71.1	60.1	60.2	55.0	64.8
	GLane3D-Large	Swin-B	C	66.0	61.1	72.5	64.2	60.1	58.0	66.9
	GLane3D-Fusion	ResNet-50	C+L	66.6	**61.7**	**72.7**	63.8	62.0	57.9	**67.7**
0.5m	PersFormer [2]	EffNet-B7	C	36.5	26.8	36.9	33.9	34.0	28.5	37.4
	Anchor3DLane [8]	EffNet-B3	C	34.9	28.3	31.8	30.7	32.2	29.9	33.9
	M^2-3DLaneNet [21]	EffNet-B7	C+L	48.2	40.7	48.2	49.8	46.2	38.7	44.2
	PersFormer [2]	ResNet-50	C	43.2	36.3	42.4	45.4	39.3	32.9	41.7
	LATR [22]	ResNet-50	C	54.0	44.9	56.2	47.6	46.2	45.5	55.6
	DV-3DLane [23]	ResNet-34	C+L	63.5	58.6	69.3	62.4	59.9	53.9	59.3
	DV-3DLane [23]	ResNet-50	C+L	65.2	59.1	69.2	**63.0**	**62.0**	**56.9**	60.5
	Glane3D-Lite	ResNet-18	C	53.8	46.7	57.7	47.9	47.1	45.8	55.7
	GLane3D-Base	ResNet-50	C	57.9	51.0	61.7	53.5	53.8	49.4	60.5
	GLane3D-Large	Swin-B	C	61.1	54.2	64.5	56.8	55.2	53.6	63.3
	GLane3D-Fusion	ResNet-50	C+L	**65.6**	**62.4**	**71.6**	62.9	61.1	**56.9**	**66.6**

Table 11. Extended Quantitative Results by Category on the OpenLane Dataset [2].

Lift Splat Shoot, on the other hand, estimates depth from frontal view features $\mathbf{F_{FV}} \in \mathbb{R}^{H' \times W' \times C}$, which is extracted from frontal view image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$. Each feature vector is projected onto the corresponding BEV grid, which is calculated based on the camera position and depth estimation. Since multiple pixels may fall within the same BEV grid, LSS applies a cumulative sum trick to pool the features that fall within each grid.

A key challenge with IPM is its reliance on projections sampled from the ground surface, which can lead to inaccurate height estimations, particularly on non-flat surfaces. To compare the two projection methods, we trained our model using both LSS and IPM, with only the projection block differing between the networks, while the rest of the architecture remained the same. The results, as shown in Tab. 10, indicate that there is negligible difference between the two projection methods in terms of F1-score. Since IPM requires less effort for deployment across different platforms, we opted to use IPM for our training.

13. Extended OpenLane Results

Category-based F1 scores with thresholds of 1.5m and 0.5m are presented in Tab. 11. The results indicate that camera-only GLane3D-Large achieves performance similar to that of camera + LiDAR fusion models in terms of F1 score at the 1.5m threshold. However, the differences in F1 scores between camera-only models and camera + LiDAR fusion models at the 0.5m threshold highlight the contribution of LiDAR to localization accuracy.

GLane3D-Fusion outperforms other models in most categories, as shown in Tab. 11. It is important to note that GLane3D-Fusion utilizes a simpler fusion approach for combining LiDAR and camera features, in contrast to the more complex fusion strategy used by DV-3DLane [23].

14. Extended Apollo Results

Tab. 12 shows the extended results in Apollo [5] dataset with previous methods.

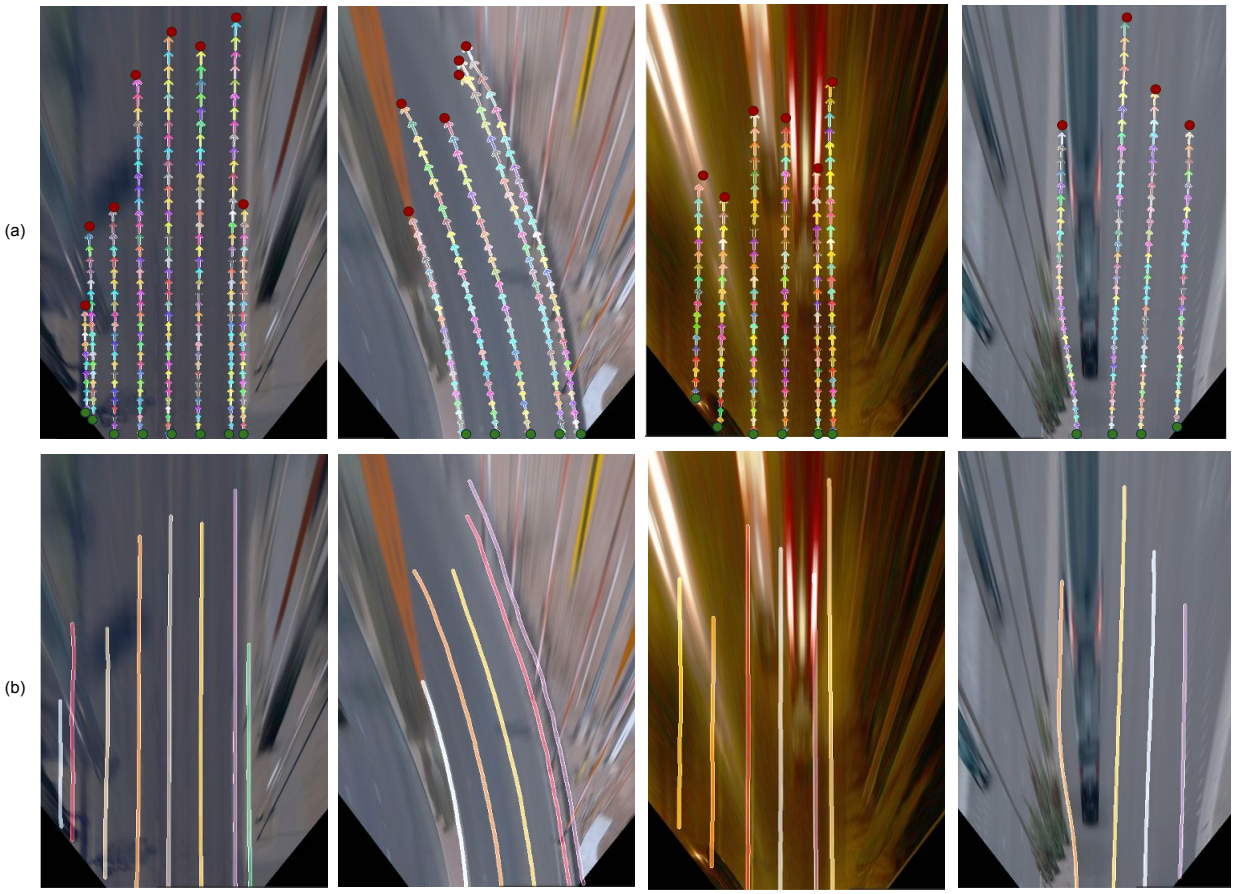

Figure 8. Lane Instance Extraction: (a) Graph of keypoints, (b) Extracted lane instances.

Figure 9. Qualitative results on cross dataset evaluation on Apollo validation set of Balanced Scenes. The rows (a), (b), (c) illustrate prediction from PersFormer[2], LATR[22] and GLane3D with 2D projection, respectively.

Subset	Methods	Backbone	F1-Score(%)↑	AP(%)↑	X-error near(m) ↓	X-error far(m) ↓	Z-error near(m) ↓	Z-error far(m) ↓
Balanced Scene	PersFormer[2]	EffNet-B7	92.9	-	0.054	0.356	0.010	0.234
	BEVLaneDet [42]	ResNet-34	96.9	-	**0.016**	**0.242**	0.020	0.216
	LaneCPP [28]	EffNet-B7	97.4	**99.5**	0.030	0.277	0.011	**0.206**
	LATR [22]	ResNet-50	96.8	97.9	0.022	0.253	**0.007**	**0.202**
	DV-3DLane [23]	ResNet-50	96.4	97.6	0.046	0.299	0.016	0.213
	GLane3D (Ours)	ResNet-50	**98.1**	98.8	0.021	**0.250**	0.007	0.213
Rare Scene	PersFormer[2]	EffNet-B7	87.5	-	0.107	0.782	0.024	0.602
	BEVLaneDet [42]	ResNet-34	97.6	-	**0.031**	**0.594**	0.040	0.556
	LaneCPP [28]	EffNet-B7	96.2	98.6	0.073	0.651	0.023	0.543
	LATR [22]	ResNet-50	96.1	97.3	0.050	0.600	**0.015**	**0.532**
	DV-3DLane [23]	ResNet-50	95.5	97.2	0.071	0.664	0.025	0.568
	GLane3D (Ours)	ResNet-50	**98.4**	**99.1**	0.044	0.621	0.023	0.566
Visual Variations	PersFormer[2]	EffNet-B7	89.6	-	0.074	0.430	**0.015**	0.266
	BEVLaneDet [42]	ResNet-34	95.0	-	**0.027**	0.320	0.031	0.256
	LaneCPP [28]	EffNet-B7	90.4	93.7	0.054	0.327	0.020	**0.222**
	LATR [22]	ResNet-50	**95.1**	**96.6**	0.045	**0.315**	0.016	0.228
	DV-3DLane [23]	ResNet-50	91.3	93.4	0.095	0.417	0.040	0.320
	GLane3D (Ours)	ResNet-50	92.7	94.8	0.046	0.364	0.020	0.317

Table 12. Extended Quantitative Results on the Apollo 3D Synthetic Dataset [5].