

## Appendix for CLRNet: Cross Layer Refinement Network for Lane Detection

Tu Zheng<sup>1,2\*</sup> Yifei Huang<sup>1,2\*</sup> Yang Liu<sup>1,2</sup> Wenjian Tang<sup>1</sup> Zheng Yang<sup>1</sup>  
Deng Cai<sup>1,2</sup> Xiaofei He<sup>1,2</sup>

<sup>1</sup>Fabu Inc., Hangzhou, China

<sup>2</sup>State Key Lab of CAD&CG, Zhejiang University, China

### A. Discussion about Lane Representation

In this appendix, we discuss different representations of lane, including mask, line, and parameter. Finally, we will explain how we choose the representation of our lane prior.

Most lane detection datasets like CULane [9], Tusimple [15], and LLAMAS [1] use equally-spaced 2D-points as lane representation. Specifically, lane is expressed as a sequence of points, *i.e.*,  $P = \{(x_1, y_1), \dots, (x_N, y_N)\}$ . The  $y$  coordinate of points is equally sampled through image vertically, *i.e.*,  $y_i = \frac{H}{N-1} * i$ , then each  $x_i$  can be associated with the corresponding  $y_i$ . Below, we will discuss three representations of the lane.

**Mask representation.** In mask representation, the lane is represented by a binary mask (*e.g.*, 0 for background, 1 for foreground). In segmentation based methods [9, 18], they predict a probability map ( $M \in \mathbb{R}^{H \times W}$ ) for each lane instance. To extract the lane object from  $M$ , they search the highest score  $x_i$  for each predefined  $y_i$ , which can be written as:

$$x_i = \operatorname{argmax}(M[y_i]).$$

These coordinates are then connected by cubic splines. This mask representation of lane is redundant since lane pixels are far fewer than mask pixels.

**Line representation.** For line representation, lanes are directly represented by discrete points, *i.e.*,  $\{(x_i, y_i) | i = 1, \dots, N\}$ . Therefore, the parameter of the line is fewer than the mask. Since the  $y$ -coordinate is predefined, a lane can then be defined only by its  $x$ -coordinate ( $X = \{x_i\}_{i=1}^N$ ). Hence, we can predict the  $X$  by classification or regression. The classification methods are UFLD [10] and CondlaneNet [6]. UFLD directly performs classification to get  $X$ . While CondlaneNet further adds an offset map to regress more precise location of  $x$ -coordinates ( $X$ ). The regression methods are Line-CNN [5] and LaneATT [13]. They are more convenient since they directly regress the precise  $x$ -coordinates ( $X$ ).

Method	Gather Module	mF1	F1@50	F1@75	F1@90
LaneATT	None	45.45	74.33	48.14	10.77
LaneATT	self-attention	47.35	75.09	51.29	12.32
LaneATT	ROIGather	<b>49.47</b>	<b>75.81</b>	<b>54.62</b>	<b>14.13</b>
CLRNet	None	54.74	78.91	61.77	20.09
CLRNet	ROIGather	<b>55.23</b>	<b>79.58</b>	<b>62.21</b>	<b>20.64</b>

Table 1. Effectiveness of the ROIGather module. “None” means remove the gather module. “self-attention” is the attention mechanism in LaneATT [13]. Models are trained/tested on the CULane dataset using ResNet18 backbone.

**Parameter representation.** The parameter representation further simplifies the lane representation, *e.g.*, PolyLaneNet [14] and LSTR [7] use a cubic curve to represent a lane line. Parameter-based methods have fewer parameters to regress, but they are sensitive to the predicted parameters, *e.g.*, the small error prediction on high-order coefficient may cause shape change of lanes. Currently, these parameter-based lane detectors still struggle to achieve high accuracy.

In our method, we choose the line representation. To be more specific, we directly regress the  $x$ -coordinates. It can have fewer parameters than mask representation. In the meantime, it is simple and more stable than the parameter representation in real detectors.

### B. More ablation results

#### B.1. ROIGather

ROIGather module can also be plugged into other networks for gathering global context. In this appendix, we study the effectiveness of ROIGather for other line anchor-based methods, *e.g.*, Line-CNN [5] or LaneATT [13]. Since only LaneATT is available for the source code, we adopt further experiments with LaneATT. We start by briefly reviewing LaneATT. LaneATT extends Line-CNN by adopting a novel self-attention [16, 17] mechanism. It performs self-attention on local lane features to gather global information. As shown in Table 1, the self-attention improves

mF1 from 45.45 to 47.35, which indicates global information is critical for detecting lanes accurately. Compared with self-attention, our ROIgather has two advantages: 1) building relations between the lane prior pixels. 2) gathering global information from the whole feature map instead of local lanes. Replacing self-attention by our ROIgather in LaneATT, the improvement is more significant, 4 points mF1 and 1.5 points F1@50. We can also enjoy considerable accuracy gains in CLRNNet. The benefits of ROIgather are witnessed for all evaluation metrics. This again demonstrates ROIgather is effective for accurate lane detection.

## B.2. Line IoU loss

In this section, we further ablate the Line IoU loss. We will discuss the extended radius  $e$  in the Line IoU loss. Moreover, we apply it to other detectors like LaneATT and we provide the qualitative results on the real dataset.

**Radius  $e$ .** First, we test the performance of LIoU with different radius  $e$ . The radius is the extended length of each point in the lane prior. This experiment is conducted on the CULane dataset using the ResNet18 [3] backbone. Results in Table 2 show that the performance first increases and then drops slightly. Since the peak performance occurs when  $e = 15$ , we set  $e = 15$  for better performance.

**Performance on real detectors.** To demonstrate the effectiveness of our Line IoU loss, we apply it to the LaneATT by replacing the smooth- $l_1$  with Line IoU loss. The result is shown in Table 3. Line IoU loss improves the mF1 by 0.6%. To be more specific, most of the improvements are from F1@75, F1@90, which yields 1 point higher F1 compared with the smooth- $l_1$  setting. Moreover, we adopt the Line IoU loss in LaneATT with ROIgather, which shows ROIgather can work together with the Line IoU loss. It improves 0.8% F1@75 and 1.3% F1@90, respectively. Furthermore, the performance on CLRNNet gets consistent improvement. These experimental results indicate that our proposed Line IoU loss dramatically improves the performance by regressing lanes as the whole unit.

**Visualization.** We provide the visualization results of smooth- $l_1$  loss and Line IoU loss in Fig. 1. In the smooth- $l_1$  loss, the coordinates of a lane are optimized as independent variables. Thus, it results in failure cases which some points are very close to the ground truth but the entire lane is unsatisfactory. On the contrary, Line IoU loss considers the lane as a unit and benefits the evaluation metric [12]. Adopting the Line IoU loss could yield a more accurate localization result, which shows the superiority of Line IoU loss for improving lane detection performance.

Radius $e$	mF1	F1@50	F1@75	F1@90
7.5	55.08	79.41	62.09	20.55
15	<b>55.23</b>	<b>79.58</b>	<b>62.21</b>	<b>20.64</b>
20	54.86	79.50	62.01	19.77

Table 2. Performance of LIoU in CLRNNet with different extend radius  $e$ . Models are trained/tested on the CULane dataset using ResNet18 backbone.

Method	LIoU	mF1	F1@50	F1@75	F1@90
LaneATT		47.35	<b>75.09</b>	51.29	12.32
LaneATT	✓	<b>47.95</b>	75.08	<b>52.25</b>	<b>13.31</b>
LaneATT + ROIgather		49.47	<b>75.81</b>	54.62	14.13
LaneATT + ROIgather	✓	<b>50.12</b>	75.77	<b>55.42</b>	<b>15.47</b>
CLRNNet		54.22	79.05	61.13	19.04
CLRNNet	✓	<b>55.23</b>	<b>79.58</b>	<b>62.21</b>	<b>20.64</b>

Table 3. Effectiveness of the LIoU loss. Models are trained/tested on the CULane dataset using the ResNet18 backbone for LaneATT and CLRNNet.

Prior number	mF1	F1@50	F1@75	F1@90	FPS
1000	<b>53.17</b>	<b>78.65</b>	<b>59.76</b>	<b>17.01</b>	150.8
300	52.70	78.32	59.51	15.97	199.7
192	52.80	78.27	59.50	16.54	213.7
128	52.39	78.14	58.96	16.05	218.3
48	51.26	77.48	57.52	14.59	223.9
32	50.67	76.99	56.49	14.74	<b>225.2</b>

Table 4. Ablation studies of on number of lane priors. Using the refinement  $R_0$  and results are reported on CULane.

## B.3. Lane priors number.

Being efficient while keeping high accuracy is crucial for a lane detection model. In this experiment, we study the number of lane priors with refinement  $R_0$ . The results are shown in Table 4. As the number of prior decreases, the F1 score drops while a great improvement in terms of efficiency. To strike a balance between speed and accuracy, we select the number of lane priors as 192.

## C. Generalization Study

In this section, we study the generalization of our proposed method. To this end, we employ the checkpoint trained from the CULane training set to perform testing with the Tusimple testing set, which is inspired by [11]. We further conduct the experiments on some recent works, *i.e.*, LaneATT [13] and CondLaneNet [6]. For a fairer comparison, we also turn parameters (*e.g.*, confidence threshold) to find the best model in the testing set. Results are shown in Table 5. The FOLOLane shows the promising generalization ability, *e.g.*, it achieves the 84.36% Accuracy even



Figure 1. Visualization of detection results. The first row is the smooth- $l_1$  loss and the second row is the Line IoU loss. The yellow line is the ground truth. In the smooth- $l_1$  loss, the coordinates of a lane are optimized as independent variables. Thus, it results in failure cases which some points are very close to the ground truth but the entire lane is unsatisfactory. On the contrary, Line IoU loss considers the lane as a unit, yielding a more accurate localization result. Best view with 300% zoom.

Method	Backbone	Accuracy (%)
SCNN [9]	VGG16	0.29
UFLD [10]	ResNet18	65.53
SIM-CycleGAN [8]	ERFNet	62.58
PINet [4]	Hourglass	36.31
LaneATT [13]	ResNet18	67.06
LaneATT [13]	ResNet34	72.89
LaneATT [13]	ResNet122	56.37
FOLOLane [11]	ERFNet	84.36
CondLaneNet [6]	ResNet18	79.91
CondLaneNet [6]	ResNet34	80.63
CondLaneNet [6]	ResNet101	80.84
<b>CLRNet (ours)</b>	ResNet18	88.17
<b>CLRNet (ours)</b>	ResNet34	<b>90.08</b>
<b>CLRNet (ours)</b>	ResNet101	89.34

Table 5. Generalization study of lane detection methods. Models are trained with the CULane dataset and tested on the Tusimple dataset. Some results are from [11].

higher than CondLaneNet. Our method outperforms other methods by a large margin (more than 5%) with an accuracy of 90.08%. This demonstrates our method has a strong generalization ability.

## D. More visual results

Fig. 2 shows the visualization result in Tusimple dataset. The lane point is predicted separately in UFLD, thus it cannot predict the smoothness lanes in some cases. Condlane

is easy to miss some lane instances since it only predicts the start point as proposals. Our CLRNet is capable of precisely detecting these lanes and providing accurate lane lines. Moreover, the last example shows CLRNet can also predict accurate lane curves in the remote part of the image.

We provide some failure cases in Fig. 3. As we discuss in Limitation Section, these failure cases are common in many detectors, *i.e.*, the remote part of the lane curve is hard to be traced. However, introducing the curve lane prior or using height-driven attention [2] may improve results, which we will study in the future.

## E. Limitations

While our method is robust in many challenging scenarios (*e.g.*, severe occlusion, extreme lighting condition), we do observe failure cases as shown in Fig. 3. The remote part of the lane curve is hard to be traced, which is also a common limitation in many anchor-based detectors (*e.g.*, CondLaneNet, LaneATT). Our work uses lane priors as our initialization lanes, which assumes the lane is a straight line and then regresses the offset to the straight line. Therefore, lane curves might not be easy to regress. These lane curves occur on the remote part of the image, which makes them have fewer visual features and hard to recognize. This limitation might be improved by introducing curve lane priors or using the height-driven attention [2] method to focus more on remote parts.

## References

- [1] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane marker dataset generation using maps. In *International*

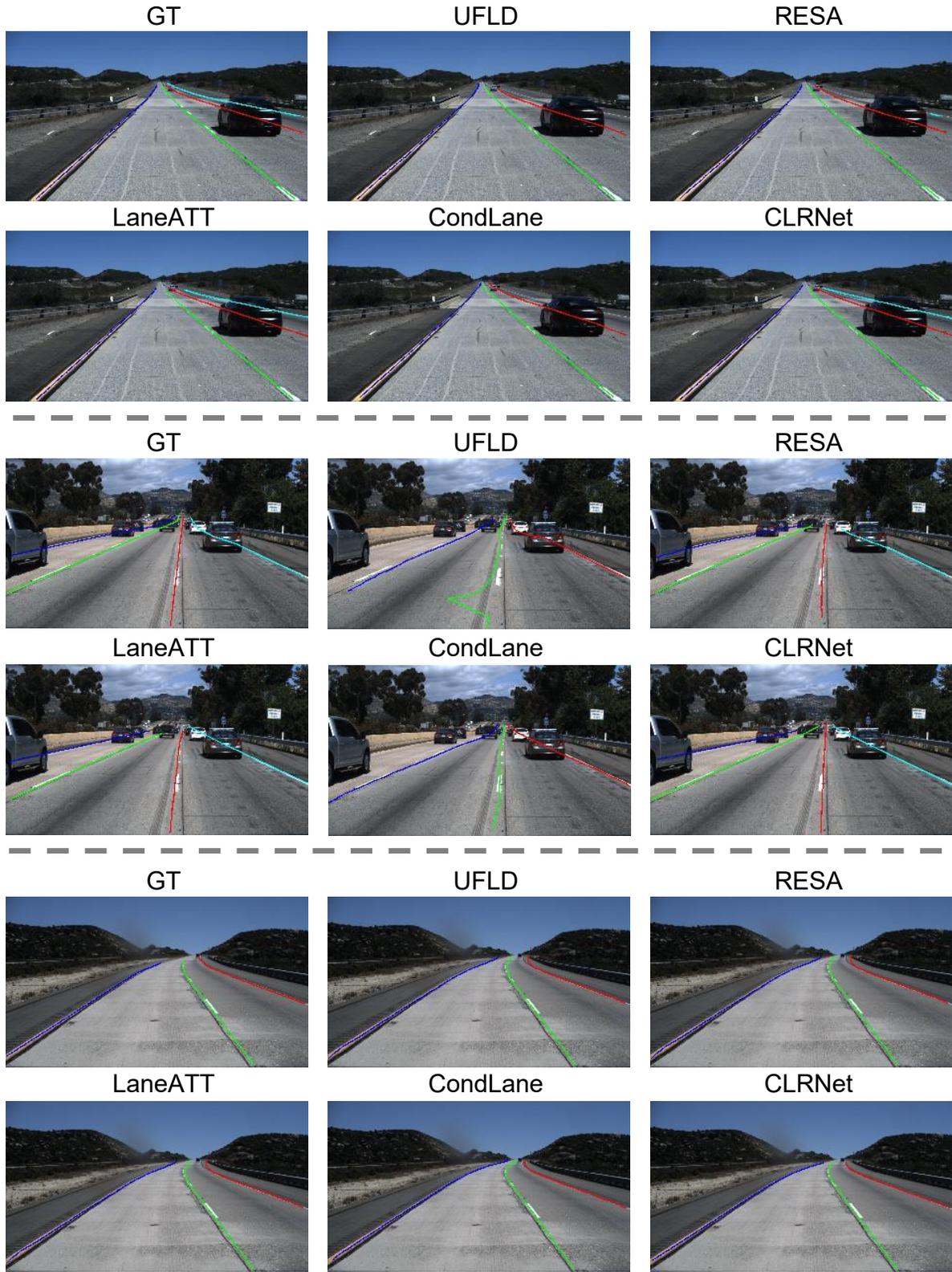


Figure 2. Visualization results on Tusimple. Our CLRNet is capable of precisely detecting lanes and providing accurate lane lines. Moreover, the last example shows CLRNet can also predict accurate lane curves in the remote part of the image. Better view with zoom.

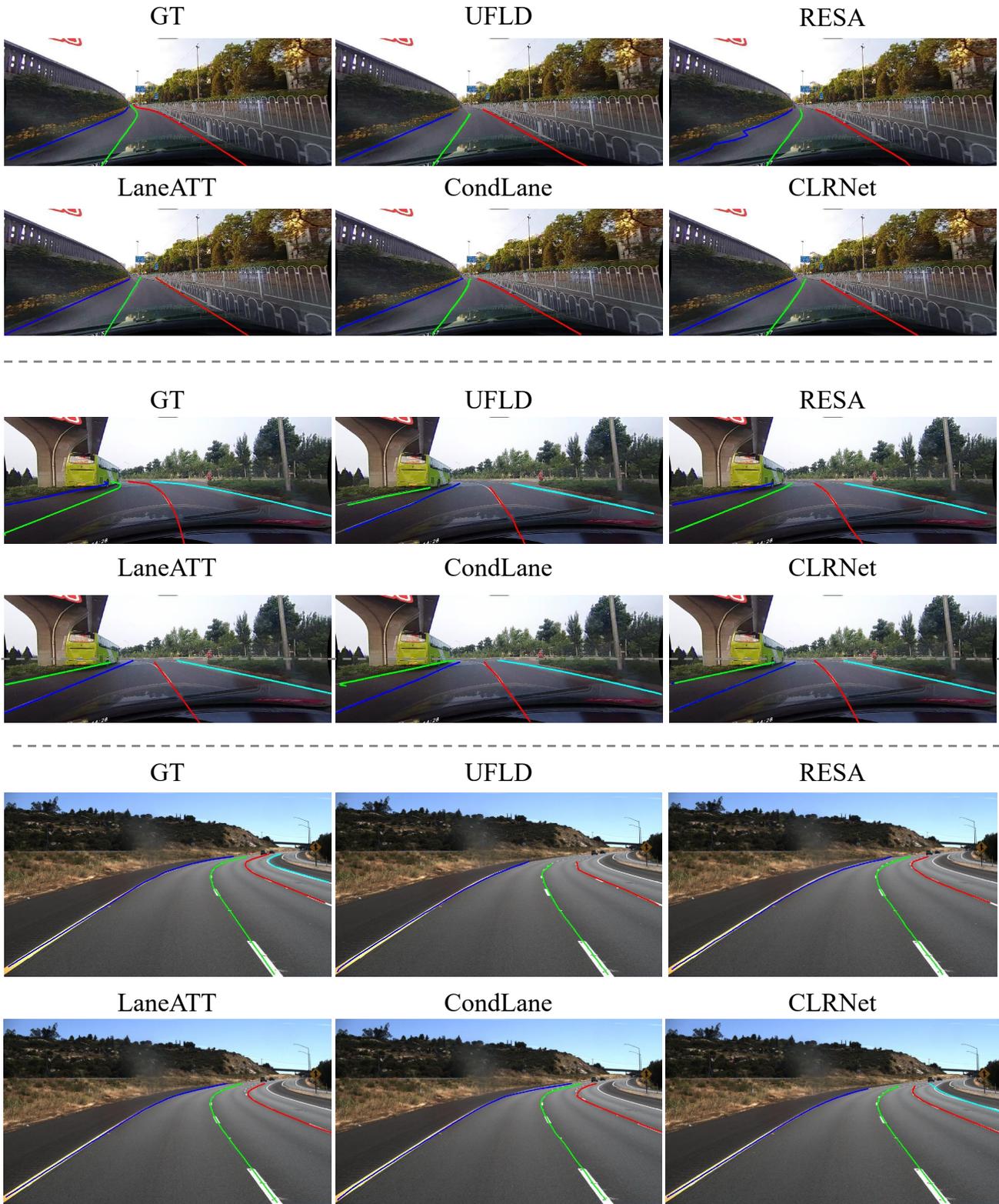


Figure 3. Illustration of some failure cases. The remote part of the lane curve is hard to be traced in these cases. Better view with zoom.

- Conference on Computer Vision (ICCV)*, volume 1, page 7, 2019. 1
- [2] Sungha Choi, Joanne T Kim, and Jaegul Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9373–9383, 2020. 3
- [3] 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. 2
- [4] Yeongmin Ko, Younkwan Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 3
- [5] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258, 2019. 1
- [6] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: a top-to-down lane detection framework based on conditional convolution. *arXiv preprint arXiv:2105.05003*, 2021. 1, 2, 3
- [7] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3694–3702, 2021. 1
- [8] Tong Liu, Zhaowei Chen, Yi Yang, Zehao Wu, and Haowei Li. Lane detection in low-light conditions using an efficient data enhancement: Light conditions style transfer. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1394–1399. IEEE, 2020. 3
- [9] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 3
- [10] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020. 1, 3
- [11] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on local: Detecting lane marker from bottom up via key point. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14122–14130, 2021. 2, 3
- [12] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 2
- [13] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 294–302, 2021. 1, 2, 3
- [14] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021. 1
- [15] TuSimple. Tusimple benchmark. <https://github.com/TuSimple/tusimple-benchmark/>, Accessed September, 2020. 1
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1
- [17] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1
- [18] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3547–3554, 2021. 1