Supplementary material
ADNet: Lane Shape Prediction via Anchor Decomposition

A. Representation of lane and lane anchor

On an image, a lane can be represented by the x-coordinates of 2D points that are equidistantly sliced by the y-axis [3]. We denote a lane as $l = \{x_1, x_2, ..., x_k\}$, where x_i represents the x-coordinate of the i-th slice and k is the total number of slices. Since the slicing scheme is fixed and equidistant (from bottom to top), the y-coordinates are distributed on a fixed pattern. Therefore, a set of y-coordinates for every lane on the image can be generated as $Y = \{y_1, y_2, ..., y_k\}$. With l and Y, we can locate the positions of points that construct a lane.

A lane anchor can be noted as (s_x, s_y, θ), where s_x and s_y represent the x-coordinates and y-coordinates of the start point, respectively, and θ represents the angle of the start point. With (s_x, s_y, θ), we can describe lane anchor as a ray using:

$$x_i = s_x \cdot w + \frac{(y_i - s_y)h}{tan(1-\theta) \cdot \pi}, \; y_i \in Y, \tag{1}$$

where w and h are the width and height of the image, θ is the angle between the anchor and the x-axis in a clockwise direction.

Table 1: Effectiveness of different MSA designs. Models are trained and tested on CULane [4] using the ResNet34 as backbone.

MSA Design	F1@50(%)
without any attention module	78.52
baseline	78.66
MSA-A: Replace $DConv_{11\times11}$ with multi-channel strip $DConv$	78.46
MSA-B: Add indentical forward path	78.34
MSA-C: Replace Liner with $DConv_{5\times5}$	**78.94**

B. Micro design of LKA

To best take advantage of a large kernel, we mainly focus on discovering an effective way to generate the attention matrix Att for the lane detection task. We demonstrate our design by gradually reconstructing the baseline (originating from [2]) to our final LKA. Apart from the baseline, Figure 2 presents our three candidate designs of the MSA module on LKA. MSA-A (Figure 2(b)) replaces the $DConv$ on the baseline with a multi-channel strip $DConv$, MSA-B (Figure 2(c)) further adds an identical forward path, and MSA-C (Figure 2(d)) replaces Liner with $DConv$ with a kernel size of 5×5. A series of experiments are shown in Table 1. Our final mechanism, MSA-C, exhibits superior performance to the other designs.

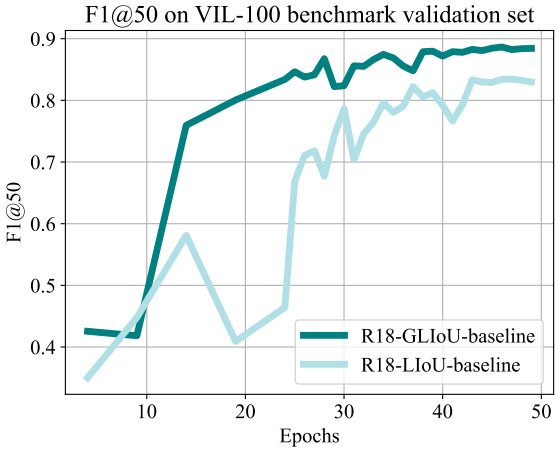


Figure 1: Illustration of the effectiveness of GLIoU on VIL-100.

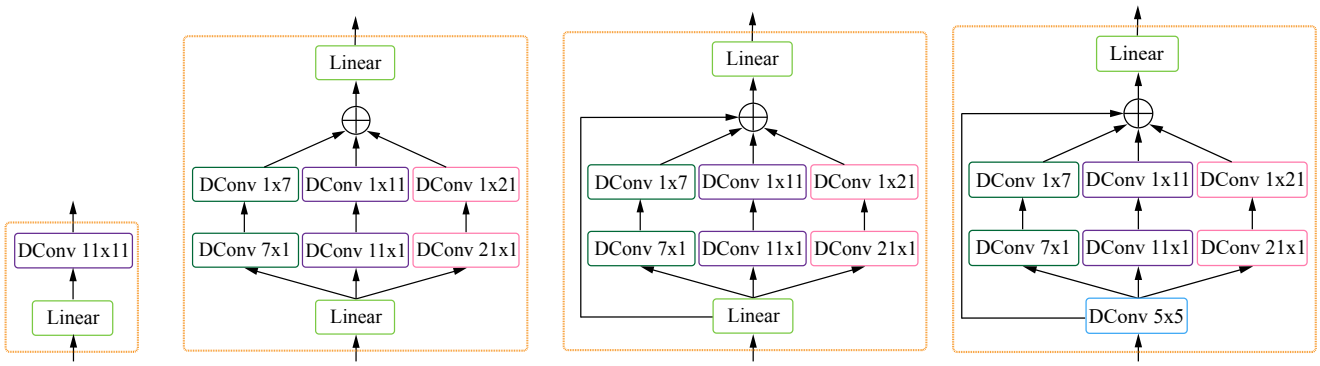
C. GLIoU

To emphasise the effectiveness of GLIoU, we present the results on the validation set of VIL-100 [5] in chart form. The validation results are presented in Figure 1. "R18-GLIoU-baseline" represents ResNet18 [1] as the backbone, GLIoU as the regression loss, and "baseline" [2] as the attention module.

The results in Figure 1 demonstrate that using GLIoU as the regression loss function generally yields better performance than LIoU [6] across all documented epochs and reaches saturation more quickly than LIoU.

D. Supervision for heat map

In our framework, we propose supervision for the start points heat map using a non-normalised Gaussian kernel. The hyperparameter σ controls the size of the region that



(a) baseline (b) MSA-A (c) MSA-B (d) MSA-C

Figure 2: Illustration of different designs of MSA.

potentially contains start points. Scenarios that typically consist of invisible start points tend to favour higher σ values compared to scenarios with clear and eye-catching start points. Figure 3 visualises the start points heat map supervision using four different σ values on CULane. On the activation map, red regions potentially contain start points, while blue regions have a lower likelihood.

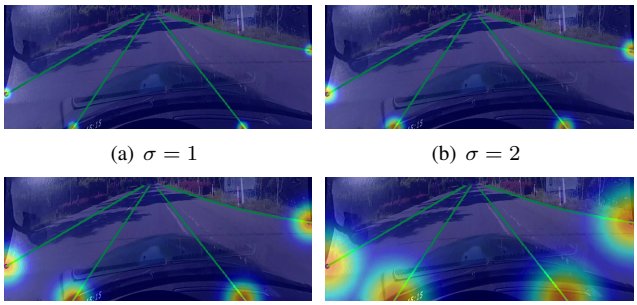
Table 2 shows the results of our experiments conducted on CULane and VIL-100. For CULane, we choose σ values of 2, 4, and 8 since start points on CULane are usually invisible. For VIL-100, we choose σ values of 1, 2, and 4 since start points are generally clear and featured in this dataset. It can be deduced that when σ on CULane is larger than 4, limited improvement can be observed, while σ lower than 4 causes a reduction in performance. VIL-100 presents an opposite tendency.

Table 2: Effectiveness of different hyperparameter σ for start points heat map supervision. Models are trained and tested on CULane and VIL-100 using the ResNet18 as backbone.

Dataset	σ	F1@50(%)	Precision(%)	Acc(%)
CULane	2	76.87	83.60	-
	4	**77.56**	85.01	-
	8	77.51	**86.25**	-
VIL-100	1	89.69	**90.24**	94.14
	2	**89.97**	90.09	**94.23**
	4	86.66	85.94	94.04

E. Number of anchors

The number of anchors can be regarded as a hyperparameter during the training and validation phases. Our experiments are conducted on CULane and VIL-100, using different backbones, ResNet18 and ResNet101. The training phase follows the procedure outlined in our paper. Results are presented in Table 3. Increasing the number of anchors does not necessarily lead to improved performance; in fact, it can even lead to a decrease. Therefore, to strike a balance across multiple indicators, the number of anchors can vary.

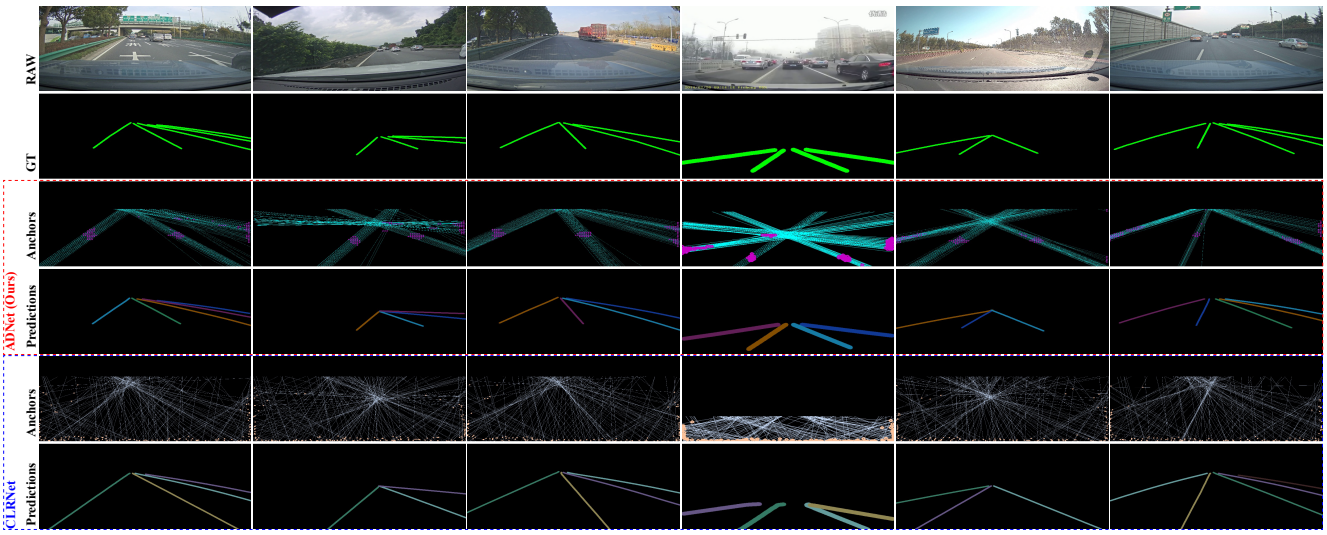


(a) $\sigma = 1$ (b) $\sigma = 2$ (c) $\sigma = 4$ (d) $\sigma = 8$

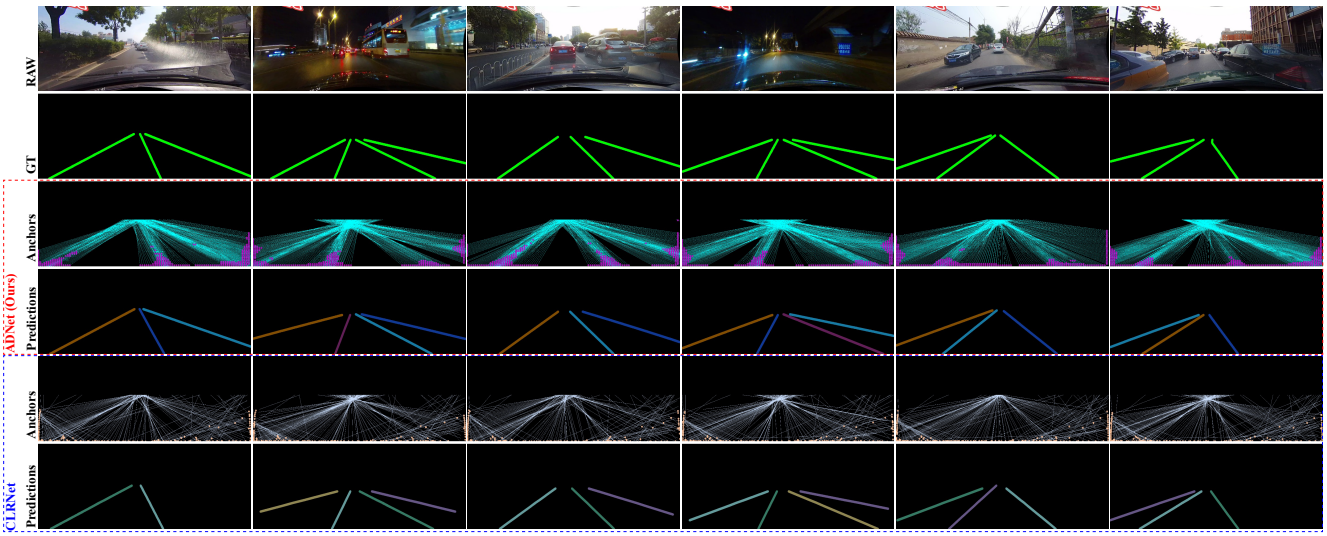
Figure 3: Illustration of start points heat map supervision with different σ on CULane.

Table 3: Effectiveness of anchors number. Models are trained and tested on CULane and VIL-100 using different backbones. "R18" stands for ResNet18.

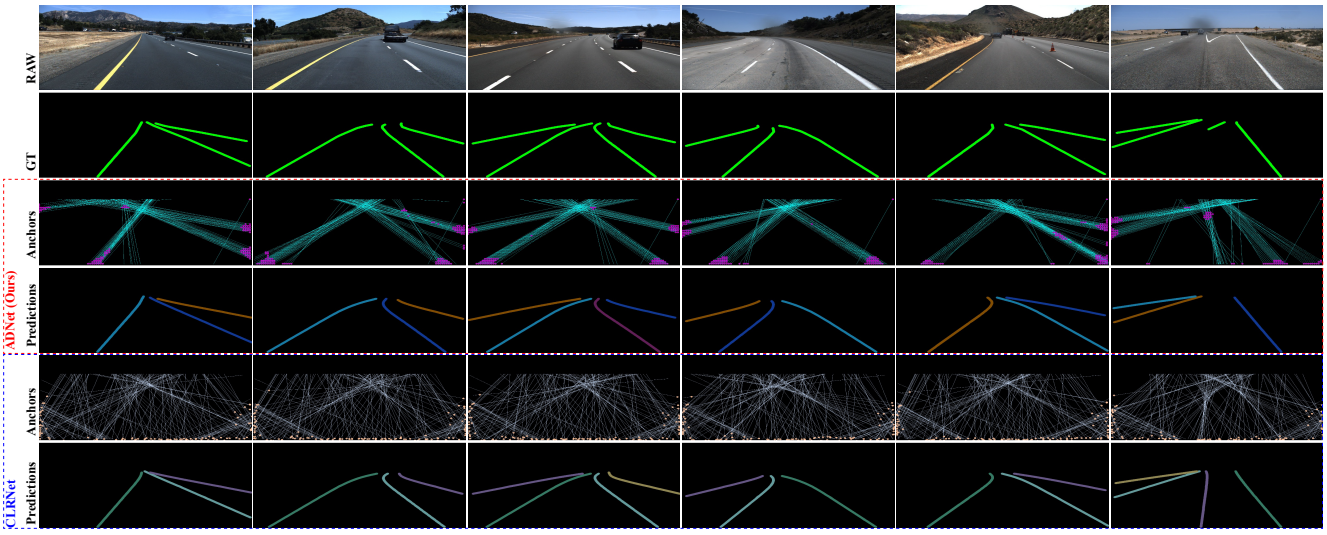
Dataset	Backbone	Anchors	F1@50(%)	Precision(%)	Acc(%)
CULane	R18	100	77.53	85.56	-
		200	77.51	85.20	-
		300	77.56	85.01	-
		400	77.63	84.85	-
		500	77.61	86.62	-
VIL-100	R18	50	90.02	90.50	94.10
		75	90.04	90.29	94.21
		100	89.97	90.09	94.23
	R101	50	90.83	91.10	94.14
		75	90.90	90.99	94.27
		100	90.90	90.99	94.27



(a) Visualisation results on VIL-100



(b) Visualisation results on CULane



(c) Visualisation results on TuSimple

Figure 4: More visualisation results on VIL-100, CULane and TuSimple compared with CLRNet [6]. Best view in zoom.

References

[1] 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. 1

[2] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022. 1

[3] 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

[4] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1

[5] Yujun Zhang, Lei Zhu, Wei Feng, Huazhu Fu, Mingqian Wang, Qingxia Li, Cheng Li, and Song Wang. Vil-100: A new dataset and a baseline model for video instance lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15681–15690, 2021. 1

[6] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 898–907, 2022. 1, 3