ADNet: Lane Shape Prediction via Anchor Decomposition

Lingyu Xiao [†], Xiang Li [‡], Sen Yang [†], Wankou Yang [†*]

[†] School of Automation, Southeast University, China
[‡] IMPlus@PCALab, VCIP, CS, Nankai University, China

lyhsiao@seu.edu.cn, xiang.li.implus@njust.edu.cn, {yangsenius, wkyang}@seu.edu.cn

Abstract

In this paper, we revisit the limitations of anchor-based lane detection methods, which have predominantly focused on fixed anchors that stem from the edges of the image, disregarding their versatility and quality. To overcome the inflexibility of anchors, we decompose them into learning the heat map of starting points and their associated directions. This decomposition removes the limitations on the starting point of anchors, making our algorithm adaptable to different lane types in various datasets. To enhance the quality of anchors, we introduce the Large Kernel Attention (LKA) for Feature Pyramid Network (FPN). This significantly increases the receptive field, which is crucial in capturing the sufficient context as lane lines typically run throughout the entire image. We have named our proposed system the Anchor Decomposition Network (ADNet). Additionally, we propose the General Lane IoU (GLIoU) loss, which significantly improves the performance of ADNet in complex scenarios. Experimental results on three widely used lane detection benchmarks, VIL-100, CULane, and TuSimple, demonstrate that our approach outperforms the state-of-the-art methods on VIL-100 and exhibits competitive accuracy on CULane and TuSimple. Code and models will be released on `https://github.com/Sephirex-X/ADNet`.

1. Introduction

Recently, the utilisation of artificial intelligence technology for the field of autonomous driving has drawn large attention from academia and industry. As a crucial part of autonomous driving system, Advance Driver Assistance System (ADAS) requires vehicles to respond timely and accurately to changes in the environment. Lane line is a vital part of the vehicle sensing the environment, as the ADAS

*Corresponding author

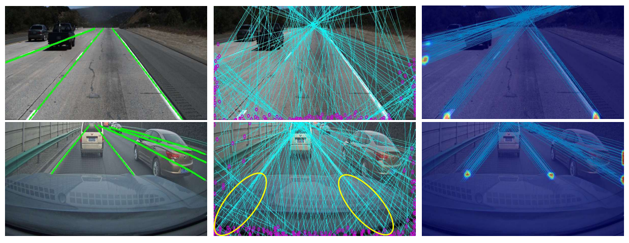

(a) Annotations (b) Traditional (c) Ours

Figure 1: Illustration of different dynamic anchor proposal methods. (a) illustrates two common lane prediction scenarios. In the first row, the lane lines originate from the edges of the image, while in the second row, the lane lines can emanate from any location within the image. (b) proposes anchor **dispersedly** [49], resulting in low anchor quality. This anchor proposal method is adequate on first row scenarios but oversimplistic on the second (emphasised by yellow oval). The points and lines represent start points and anchors respectively. (c) we propose anchor **concentratively**, possible start points are shown on activation map, anchors can merge from the whole image, which ensures anchor quality and flexibility.

needs the shape of lane lines to keep the vehicle on track.

With the advancement of CNN, recent studies on lane shape detection have made great progress on either accuracy [35, 40, 49] or real-time performance [31, 32, 24]. Anchor-based methods have shown superior accuracy and efficiency compared to other methods on popular benchmarks such as [29, 2, 37]. However there are still challenges to the wide application of anchor-based methods.

The first issue is the flexibility of anchors. Previous anchor-based methods [35, 21, 49, 31] have posited that lane lines can only originate from the three edges of an image (left, bottom, right). While this assumption leverages prior information of lane lines to achieve favourable accuracy and speed, it is oversimplistic as lane lines do not always start from the three edges due to the obstructions such as vehicles in adjacent lanes or a vehicle's front hood (shown on Figure 1(a)).

Another problem is the low quality of anchors. Anchor-based methods [21, 35] usually employ an approach of fixed anchors, while recent method [49] adopt a dynamic approach with dispersed anchor prediction. This dispersed prediction (shown on Figure 1(b)) is possibly unreliable when the camera resolution is varying, particularly in cases where lane lines do not start from the edges. Additionally, the inherent physical characteristics of lane lines, such as their slenderness and continuity, presenting significant challenges in capturing their geometric features. However, most existing approaches are limited in small kernel sizes which present an obstacle to accurately capture the whole feature descriptors of lane lines.

In this paper, to address these problems, we introduce Anchor Decomposition Network (ADNet). Specifically, to make anchors flexible, we propose Start Point Generate Unit (SPGU) which decomposes them into predicting the position of the start points and its associated direction on a global scale by the probability map (heat map). To enhance anchors' quality, we realise the crucial role of large receptive in capturing slender and continue lane lines. Therefore we introduce a Large Kernel Attention (LKA) module and integrate it with the Feature Pyramid Network (FPN). Since we predict anchors in a concentrative way (shown in Figure 1(c)), the location is invariant to the density of validated pixels and thus the anchors' quality and flexibility can be mutually guaranteed.

We conduct extensive experiments on three lane detection benchmarks: VIL-100 [47], CULane [29] and TuSimple [37]. Comparing along with state-of-art methods, our approach shows excellent performance on all datasets. In particular, on VIL-100 dataset where segmentation-based methods always perform superior to the anchor/keypoint-based counterparts, our framework outperforms all existing state-of-art methods, making the anchor-based method a more generalised pipeline. The main contributions of this paper are summarised as :

- We emphasise the importance of anchor flexibility for the anchor-based approaches by explicitly decomposed learning of the heat map of starting points and their associated directions. The decomposition makes our algorithm adaptable to different lane types in more scenarios.

- To our best knowledge, we are the first to investigate the effectiveness of Large Kernel mechanism on lane detection task to guarantee the anchor quality, as the lane lines usually cover the entire image which often require significantly large context to be accurately captured.

- We rethink the limitations of LIoU loss and propose our own General Line IoU (GLIoU) loss tailor for anchor-based lane detection method on complex scenarios. Furthermore, we utilise the explicit physical modelling by anchor decomposition to guide the learning of kernel offsets in the proposed Adaptive Lane Aware Unit (ALAU).

- Experiments on main benchmarks show excellent trade-offs on performance and speed compared with SOTA methods, outperforming all recent methods on VIL-100 dataset.

2. Related Work

2.1. Segmentation-based methods

In segmentation-based methods, the task of identifying lane lines has been converted to a per-pixel prediction task. [29] first introduces a spatial mechanism passing messages between pixels row-wise and column-wise that fails to perform in real time. [48] further proposes a recurrent aggregator fully utilised lane shape priors to obtain better performance. On [1], additional affinity fields are predicted simultaneously with the binary segmentation map, which is used in the decoder to cluster lane pixels. Segmentation-based method can achieve high accuracy when lane lines are visible, but it's unstable in complex traffic scenarios and inefficient.

2.2. Anchor-based methods

Anchor-based & detection-based methods define lane lines in a similar way. They divide an image into slices or cells, and then convert the lane detection task into either offsets' regression on each slice or a row-wise classification task. [31] first predicts lane lines via a simple linear layer using row-wise classification. [21] improves the representation of lane lines by converting cell representation into anchor representation, and identifying lane shape through regression of the offsets on every slice between anchors and ground truth. [35] further enhances this formulation by adding anchor-based pooling and a lane attention mechanism to it. [32] proposes a hybrid anchor system to improve the performance of UFLD. [24] proposes a conditional convolution and RIM migration to solve the instance-level discrimination problem on lane detection. [49] develops ROIGather to fuse lane context from different layers and, for the first time, changes the anchor-based formulation into an anchor-free manner, achieving state-of-the-art performance on multiple benchmarks.

Anchor-based and detection-based methods heavily rely on the position of anchors. On one hand, this can bring higher accuracy since anchors contain prior information on lane lines. On the other hand, these inherent properties lead

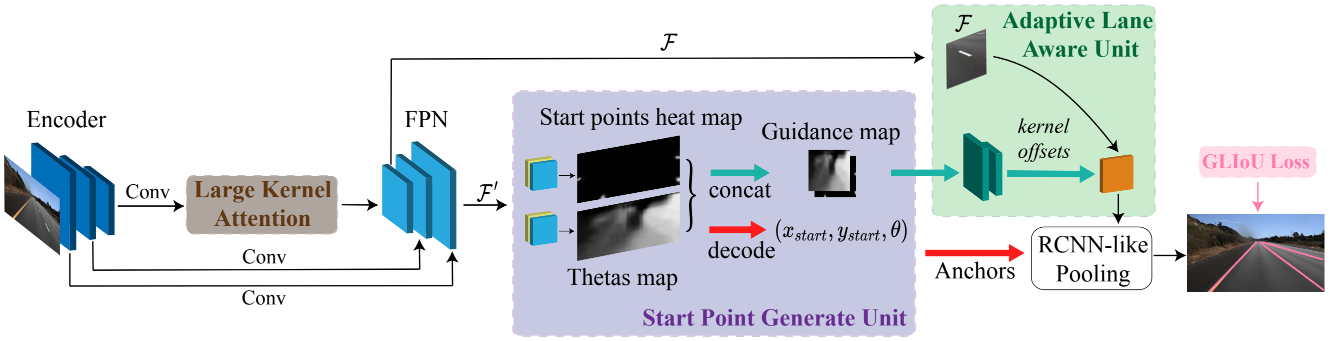


Figure 2: Overview of our ADNet. Lane context first extracted by the encoder and enhanced by FPN embedded with Large Kernel Attention (LKA), which plants after FPN's lateral layer to reduce computation cost. Then, low-level context $\mathcal{F}'$ is delivered into Start Point Generate Unit (SPGU) to generate start point guided anchors and guidance map, while high-level context $\mathcal{F}$ is further aggregated through Adaptive Lane Aware Unit (ALAU) with the help of the auxiliary guidance map. After pooling, we optimise lane lines via General Lane IoU loss.

to some shortcomings, such as the starting point of the anchor may not always be located on the three edges of the image, limiting its application.

2.3. Keypoint-based methods

Keypoint-based methods treat lane lines' prediction as a key point estimation task. Usually, the algorithm will first predict all the possible key points that most likely belong to lane lines, and follow up with a post-process of assigning different points to different lanes. [19] predicts key points on lane lines and distinguishes each instance by embedding features of predicted points. [33] predicts local key points in a bottom-up manner and refiners key points' location via its offsets between adjacent points. [40] clusters points via offsets between key points and start points, and a modified deformable convolution network [4] to extract holistic lane features. Lane instances are predicted by keypoint-based methods via low-efficient post-processing of key points from the heat map, moreover, the accuracy of the algorithm highly relies on the resolution of the input image, together with time-consuming post-processing, making keypoint-based methods hard to strike a balance between latency and accuracy.

3. Approach

The overall structure is illustrated in Figure 2. Our algorithm contains four parts: Start Point Generate Unit (SPGU), Adaptive Lane Aware Unit (ALAU), General Lane IoU loss and Large Kernel Attention (LKA).

3.1. Start point generate unit

Motivation. On lane shape detection tasks, predefined anchors have direct affection toward the anchor-based & detection-based method [35, 21]. Literature like [49, 3] perform in an anchor-free manner but they work under the assumption that lane line rays from three edges of the image, therefore, limiting its application.

Structure. The ultimate goal is to form an anchor by predicting the start point location and theta given downsampled feature map $\mathcal{F}' \in \mathbb{R}^{H'_f \times W'_f}$, which can be formulated as $p(x_{start}, y_{start}, \theta|\mathcal{F}')$. Like most of the Keypoint-based detection framework [20, 50, 8, 44], we aim to predict the start point for each lane on the image by estimating the possibility of a start point on a certain region of the downsampled heat map. Additionally, we observe that the theta of the anchor is closely associated with the start point in terms of spatial relation [39], according to the Bayes' theorem, its location and shape can be decomposed as:

$$p(x_{start}, y_{start}, \theta|\mathcal{F}') = p(x_{start}, y_{start}|\mathcal{F}')p(\theta|x_{start}, y_{start}, \mathcal{F}'). \quad (1)$$

During the training phase, we generate a supervision heat map by adding a non-normalised Gaussian kernel to each ground truth start points:

$$H^{pts}_{gt}(x, y) = exp(-\frac{(x-x^{start}_{gt})^2+(y-y^{start}_{gt})^2}{2\sigma^2}) \text{ , } H_{gt} \in \mathbb{R}^{H'_f \times W'_f}, \quad (2)$$

x, y is the coordinate of pixels on H^{pts}_{gt}; $x^{start}_{gt}, y^{start}_{gt}$ is ground truth start point's coordinate; σ is a hyperparameter. Then supervision for theta map $H^{\theta}_{gt}(x, y)$ can be formulated as:

$$H^{\theta}_{gt}(x, y) = index(H^{pts}_{gt}(x, y) > t_\theta) \cdot \theta(x^{start}_{gt}, y^{start}_{gt}). \quad (3)$$

We can interpret this as follows: if the probability of start points in a particular region is greater than t_θ, we consider that region to share the same θ as the ground truth start point. Unlike a strict assumption of one-point-one-theta, our approach expands the potential occurrence area of proposal anchors, providing a wealth of high-quality anchors for regression. This allows neural networks to determine the best fit under different conditions, without needing to add an additional loss to compensate for the offset between

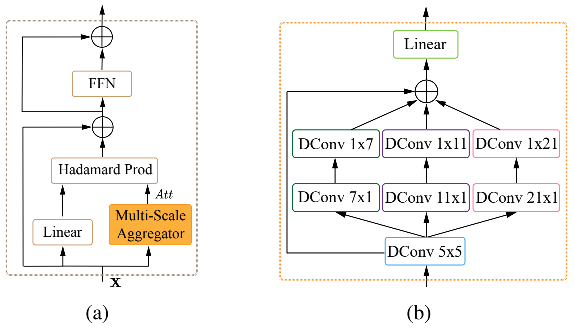


Figure 3: Illustration of LKA. LKA can be seen as the combination of (a) attention mechanism and (b) Multi-Scale Aggregator (MSA) .

points on the heat map and the original image due to downsampling [40].

We modified [23] to meet the imbalance between start point regions and the non-start point regions:

$$\mathcal{L}_{hm} = \frac{-1}{H'_f \times W'_f} \sum_{xy} \begin{cases} (1-H^{pts}_{pred})^{\alpha} log(H^{pts}_{gt}) & H^{pts}_{gt} = 1 \\ (1-H^{pts}_{gt})^{\beta} (H^{pts}_{pred})^{\alpha} log(1-H^{pts}_{pred}) & otherwise \end{cases}, \tag{4}$$

α and β are hyperparameters of focal loss. Similarly, we modified L1 loss for theta map over the whole feature map:

$$\mathcal{L}_{\theta} = \frac{1}{H'_f \times W'_f} \sum_{xy} |H^{\theta}_{pred} - H^{\theta}_{gt}|. \tag{5}$$

It is noteworthy that the calculation of theta map loss in the non-start point region is unnecessary due to the uncertainty of their theta values.

3.2. Large kernel attention

Motivation. In recent literature [6, 7], it has been observed that the performance of ConvNet is restricted when the kernel size exceeds 7×7, thereby limiting the potential benefits of mixed Transformer architecture for downstream tasks that require a large receptive field. Building on the work of [15, 11], we devise a Large Kernel Attention (LKA) module integrated with Feature Pyramid Network (FPN) specifically for lane detection.

Design. In Figure 2, it can be seen that our LKA module is placed after the lateral layer of FPN to minimise computation cost. Unlike generating a similarity score *Att* between the query and value outputs, we employ Multi-Scale Aggregator (MSA) to quantify the correlation among input tokens. The mathematical expression of our approach depicted in Figure 3 can be represented as follows:

$$Att = \mathbf{W}_1(\sum_{i=0}^{3} MultiCh_i(DConv_{5\times5}(\mathbf{X}))), \tag{6}$$

$$\mathbf{Z}_1 = Att \odot (\mathbf{W}_2\mathbf{X}) + \mathbf{X}, \tag{7}$$

$$\mathbf{Z} = FFN(\mathbf{Z}_1) + \mathbf{Z}_1. \tag{8}$$

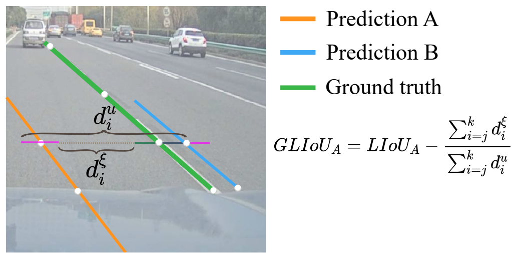

Prediction A
Prediction B
Ground truth
d_i^u
d_i^ξ
$GLIoU_A = LIoU_A - \frac{\sum_{i=j}^{k} d_i^\xi}{\sum_{i=j}^{k} d_i^u}$

Figure 4: Illustration of GLIoU. A special scenario that LIoU fails to address properly. We have extended each point on the slice to form lane segments with a certain width e, and it is evident that the L1 distance between the two segments is significantly larger than what can be captured by LIoU ($LIoU_A = -0.5, LIoU_B = 0.5, distanceL_1^A = 6e$, $distanceL_1^B = e$).

In Figure 3(b), the four feed-forward paths are denoted as $MultiCh_i$ and are distinguished by different colours, where $MultiCh_0$ corresponds to an identical forward path. Instead of using a 7×7 depth-wise convolution, strip-like convolutions are more effective in identifying lane lines while reducing computation cost. The linear layer is represented by $\mathbf{W}_i$. As suggested in [15], we use Hadamard product (denoted as $\odot$ in Eq. (7)) instead of matrix product to leverage the advantages of large kernels in MSA.

3.3. General Lane IoU loss

Motivation. Recently, LIoU [49] loss has been proposed to address the problem that the lane shape information on the anchor-based method is considered to be independent for each point when applying L1 loss. Although LIoU loss incorporates information on lane shape into a normalized metric that is invariant to scale, it may not be suitable for infrequent scenarios. Figure 4 depicts a typical scenario that exposes the limitations of LIoU. As shown in Figure 4, the LIoU values for *Prediction A-Gt* and *Prediction B-Gt* are -0.5 and 0.5, respectively, whereas the L1 distance gap between the two is significantly larger than what LIoU can capture ($distanceL_1^A = 6e$, $distanceL_1^B = e$). In other words, *Prediction A* is substantially worse than *Prediction B* according to the L1 distance metric, yet LIoU fails to account for this relationship.

Design. To overcome this limitation, we propose General Lane IoU (GLIoU), which can be considered as a generalisation of LIoU, where an additional penalty term is incorporated to highlight the spatial relationship between two lanes that do not overlap. Similar to LIoU, we begin by extending each point to form lane segments with a certain width e and computing the intersection over union ratio as usual. Then, we calculate the L1 distance between each pair of extended segments to obtain a gap distance d_i^ξ:

$$d_i^\xi = ReLU(d_i^u - 4e). \tag{9}$$

Using LIoU subtract the ratio of gap distance to the union, which is illustrated as follows:

$$GLIoU = \frac{\sum_{i=j}^{k} d_i^o - ReLU(d_i^u - 4e)}{\sum_{i=j}^{k} d_i^u}, \tag{10}$$

where j represents the index of the first validated point. On Eq. (9) and Eq. (10), d_i^u and d_i^o is defined as introduced in [49]. If the prediction overlaps with the ground truth, GLIoU degenerates into LIoU. However, if the prediction does not overlap with the ground truth, we introduce an extra penalty term $\frac{\sum_{i=j}^{k} d_i^\xi}{\sum_{i=j}^{k} d_i^u}$ to more accurately capture the L1 distance while still considering the lane as a unified entity.

The GLIoU loss can be defined as:

$$\mathcal{L}_{GLIoU} = 1 - GLIoU. \tag{11}$$

The domain of GLIoU is $(-2, 1]$, when the predicted lane perfectly matches the ground truth, GLIoU equals 1. Although the lower bound of GLIoU is -2, which may seem asymmetric, it is a more appropriate choice for predicting lane shapes since, in most cases, the prediction and ground truth do not perfectly align. Rather than emphasising the overlapped section, the GLIoU loss focuses on improving the poorly overlapped section.

3.4. Adaptive lane aware unit

Motivation. The use of traditional convolution networks for lane detection may not be optimal, as they operate on a fixed grid that does not align well with the irregular shape of lane lines. Although the Deformable Convolution Network (DCN) [4] has found extensive application in object detection, its potential for lane detection has not been fully explored [40]. Directly applying DCN to lane detection is challenging, as it is not feasible to learn kernel offsets from high-context lane features. Instead, we observe that start points and their associated thetas can be regarded as an effective guidance to predict kernel offsets due to their explicit physical modelling.

Structure. Once we have obtained the start points coordinates and theta values that are spatially related using SPGU, we encode the thetas heat map and start points heat map with dense lane information, which can be represented as $\Theta_{xy} = \{\theta_1, \theta_2, ..., \theta_N\}$ and $P_{xy} = \{p_1, p_2, ..., p_N\}$ respectively, as shown in Figure 5. For instance, we can abstract the task of predicting a set of kernel offsets on one activation unit (green dot) as follows:

$$\Delta K_{xy} = \phi(\vec{v} \cdot pts) = \phi(x, y, \theta), \tag{12}$$

$$\mathcal{S} = \{(-1,-1), (0,-1), ..., (0,1), (1,1)\}, \tag{13}$$

$$\Delta K_{xy} = \{\Delta k_{xy}^i | i = 1, ..., |\mathcal{S}|\}. \tag{14}$$

Let $\mathcal{S}$ denote the grid defined by the receptive field size and dilation, and let pts denote the position of the activation unit, with coordinates x and y. Let $\vec{v}$ be a unit vector parallel to the spatially nearest anchor, with θ denoting the anchor's theta value. Let ΔK_{xy} be the kernel offsets on pts, where ks is the kernel size and ϕ is a non-linear function. Since the theta value in non-start point regions is uncertain and the anchors' direction is mutually related, SPGU will automatically learn the theta value that has the highest probability for the region, ensuring the existence of the function $f_v : \Theta_{xy} \rightarrow \vec{v}$. Since the start point and its theta value are spatially related, the function can be further expressed as $f_v : (\Theta_{xy}, P_{xy}) \rightarrow \vec{v}$. Therefore, Eq. (12) can be rephrased as:

$$\Delta K_{xy} = \phi(f_v(\Theta_{xy}, P_{xy})). \tag{15}$$

We can simply use convolutional neural network to fit these functions. Following [4] we integrated kernel offsets with deformable convolution to adaptively extract context of activation unit, which can be expressed as:

$$\hat{\mathcal{F}}(pts) = \sum_{pts_{xy}^i \in \mathcal{S}} w(pts_{xy}^i) \cdot \mathcal{F}(pts + pts_{xy}^i + \Delta k_{xy}^i), \tag{16}$$

where $w(pts_{xy}^i)$ is the weights of convolution. In Figure 5, the green dot, blue dots, yellow arrows, and red dots correspond to pts, $pts + pts_{xy}^i$, Δk_{xy}^i, and $pts + pts_{xy}^i + \Delta k_{xy}^i$, respectively.

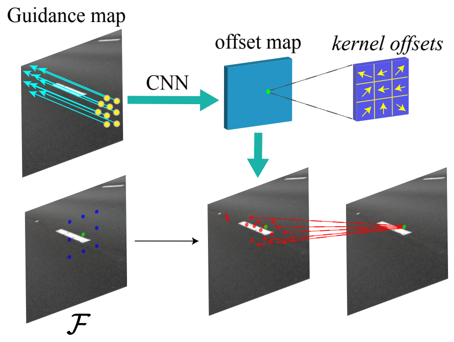


Figure 5: Illustration of ALAU. We utilise the SPGU-generated guidance map to predict kernel offsets and embed them with deformable convolution to gather lane context. In the following image, the green dot represents the activation unit, the yellow arrow indicates the offset learned from the guidance map, and the red dot denotes the sampling location ($9 \times 2 \times 1 = 18$ on each image) in a single 3×3 kernel level. The deformable group is set to 2.

3.5. Model training detail

Label assignment. Since the algorithm works in an anchor-free style, assigning a positive label in a predefined anchor manner is not feasible. We follow [49] to assign labels dynamically [10].

Loss function. The overall loss function can be written as:

$$\mathcal{L} = w_{reg}\mathcal{L}_{GLIoU} + w_{cls}\mathcal{L}_{cls} + w_{hm}\mathcal{L}_{hm} + w_\theta\mathcal{L}_\theta. \tag{17}$$

The loss function comprises four components: $\mathcal{L}_{GLIoU}$ is the General Lane IoU loss (Eq. (11)) between proposals and ground truths; $\mathcal{L}_{cls}$ is focal loss [23] between proposals and ground truths; $\mathcal{L}_{hm}$ is modified focal loss (Eq. (4)) between start points heat map and ground truth; $\mathcal{L}_{\theta}$ is modified L1 loss (Eq. (5)) between thetas heat map and ground truth. The total loss is obtained by taking the weighted sum of each component.

4. Experiments

4.1. Datasets and evaluation metric

In this paper, we use three popular benchmarks: VIL-100, CULane and Tusimple.

VIL-100 [47] is a recently released video instance lane detection dataset, that contains 10,000 frames. There are 10 scenarios in collection including multi-weather, multi-traffic scenes, day and night. The resolution of the image varies from 640×368 to 1920×1080 and lanes may locate in 8 different places, which challenges the algorithm.

CULane [29] contains 88,880 images for training. The main scene is urban traffic, which also includes various scenery, such as daytime, night, crowded, fog, etc., making it a very challenging dataset. All annotated images are 1640×590 pixels in size.

TuSimple [37] consists of simple scenes where lane lines are easily identifiable. Each annotated image has a size of 1280×720 pixels, and contains a maximum of five lane lines.

Evaluation metric. There are two main evaluation metrics widely used in lane detection: F1 and Accuracy. F1 is defined as $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$. To evaluate IoU, lanes are extended with a width of 30 pixels [29], and predictions with IoU greater than a threshold are considered as true positive (TP). Accuracy (Acc) is defined as $Acc = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}$, where C_{clip} is the number of points within 20 pixels of ground truth per image, denoted as correct points; S_{clip} is the total number of points within an image. A prediction is considered correct if it has more than 85% of points noted as correct points. False Positive Rate (FPR) and False Negative Rate (FNR) are defined as $FPR = \frac{F_{pred}}{N_{pred}}$ and $FN = \frac{M_{pred}}{N_{gt}}$, respectively.

4.2. Implementation details

We employ Resnet [14] pre-trained on ImageNet [5] as the backbone.

For the TuSimple dataset, we utilise Adam [18] optimiser with an initial learning rate of 2e-5 per batch, and train for 150 epochs using the CosineLR [26] learning rate decay strategy. The number of anchors is set to 100, and the hyperparameters for the supervision of the heat map are set to $\sigma = 2$ and $t_{\theta} = 0.2$, respectively. The weights for the loss function in Eq. (17) are set to $w_{reg} = w_{cls} = w_{hm} = 10$, and $w_{\theta} = 1$.

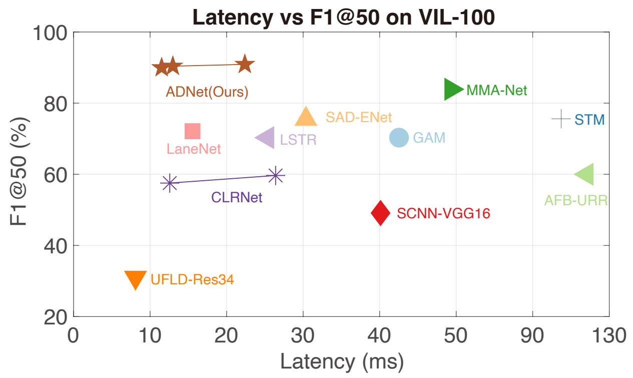


Figure 6: Latency vs F1@50 of other methods on VIL-100 lane detection benchmark. Our method outperforms all existing methods and maintains a promising inference speed.

For the CULane dataset, we use AdamW [27] optimiser and train for 15 epochs with 300 anchors. The loss function's weights are $w_{reg} = w_{cls} = 6$, $w_{hm} = 2$, and $w_{\theta} = 3$, while the hyperparameters for the heat map are $\sigma = 4$ and $t_{\theta} = 0.5$.

We use the same training settings for VIL-100 as TuSimple, except the training epoch is 80.

During training and inference, we resize input images for all datasets to 800×320. The extend radius e in GLIoU is set to 15. For FPS test on Table 1 and Table 2, we set the batch size to 1 and forward model for 2000 times. All experiments are conducted on a single RTX3090.

4.3. Performance on benchmarks

VIL-100. Our approach achieves state-of-the-art results on the recently released VIL-100 lane detection dataset. In Table 2, we compare our results with the previous state-of-the-art method MMA-Net [47], and show that our method has increased F1@50 from 83.90 to 90.90. We have also achieved a lane accuracy of 94.27 with ResNet101 and 94.38 with ResNet34, which is much better than MMA-Net. Our results have also compared with the anchor-free state-of-the-art method CLRNet, which performs very well on multiple benchmarks such as CULane and LLAMAS [2]. However, on VIL-100, CLRNet fails to maintain its edge. To provide a fair comparison with CLRNet, we have relocated our start points into three edges of the image, and our smallest model has outperformed CLRNet in multiple indicators.

CULane. The performance of ADNet is compared with other state-of-the-art methods on CULane and the results are presented in Table 1. Compared to the previous fixed anchor method, for example, LaneATT [35], our method achieves a convincing F1 score of 78.94 with ResNet34, outperforming LaneATT with ResNet122 by 1.92%. Our

Table 1: Comparison with state-of-art methods on CULane test set. 'R18' stands for ResNet18, the rest can be analogised.

Method	**F1@50**$\uparrow$	**FPS**$\uparrow$	**Normal**$\uparrow$	**Crowded** $\uparrow$	**Dazzle**$\uparrow$	**Shadow**$\uparrow$	**No Line**$\uparrow$	**Arrow**$\uparrow$	**Curve**$\uparrow$	**Cross**$\downarrow$	**Night**$\uparrow$
Segmentation Based											
SCNN-VGG16 [29]	71.60	25	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10
RESA-R50 [48]	75.30	65	92.10	73.10	69.20	72.80	47.70	88.30	70.30	1503	69.90
SAD-ENet [16]	70.80	33	90.10	68.80	60.20	65.90	41.60	84.00	65.70	1998	66.00
LaneAF-DLA34 [1]	77.41	28	91.80	75.61	71.78	**79.12**	**51.38**	86.88	**72.70**	1360	73.03
AtrousFormer-R34 [43]	**78.08**	-	**92.83**	**75.96**	**69.48**	77.86	50.15	**88.66**	71.14	**1054**	**73.74**
Keypoint Based											
PINet-Hourglass [19]	74.40	27	90.30	72.30	66.30	68.40	49.80	83.70	65.20	1427	67.70
FOLOLane-ERFNet [33]	78.80	-	92.70	77.80	**75.20**	79.30	52.10	89.00	69.40	1569	**74.50**
GANet-R34 [40]	**79.39**	69	**93.73**	**77.92**	71.64	**79.49**	**52.63**	**90.37**	**76.32**	**1368**	73.67
Parameter Based											
BézierLaneNet-R34 [9]	75.57	78	91.59	73.20	69.20	**76.74**	48.05	87.16	62.45	888	69.90
Laneformer-R50 [12]	**77.06**	-	**91.77**	**75.41**	**70.17**	75.75	**48.73**	**87.65**	**66.33**	**19**	**71.04**
Anchor & Detection Based											
FastDraw-R50 [30]	-	-	85.90	63.60	57.00	69.90	40.60	79.40	65.20	7013	57.80
UFLDv2-R34 [32]	76.00	**114**	92.50	74.80	65.50	75.50	49.20	88.80	70.10	1910	70.80
CurveLanes-L [42]	74.80	-	90.70	72.30	67.70	70.10	49.40	85.80	68.40	1746	68.90
LaneATT-R122 [35]	77.02	38	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81
SGNet-R34 [34]	77.27	-	92.07	75.41	67.75	74.31	50.90	87.97	69.65	1373	72.69
CondLane-R34 [24]	78.74	70	93.38	77.14	71.17	**79.93**	51.85	89.89	**73.88**	1387	73.92
CLRNet-R34 [49]	**79.73**	63	**93.49**	**78.06**	**74.57**	79.92	**54.01**	**90.59**	72.77	1216	**75.02**
ADNet-R18 (Ours)	77.56	87	91.92	75.81	69.39	76.21	51.75	87.71	68.84	**1133**	72.33
ADNet-R34 (Ours)	78.94	77	92.90	77.45	71.71	79.11	52.89	89.90	70.64	1499	74.78

Table 2: Comparison with state-of-art methods on VIL-100 test set. Our proposed ADNet are flexible in modelling the locations of start points (they can be anywhere in the images). For more comparisons, we also provide a "ADNet*" version where the start points are extended to the three edges likewise in CLRNet, showing inferior performance.

Methods	**F1@50**$\uparrow$	**Acc**$\uparrow$	**FP**$\downarrow$	**FN**$\downarrow$	**FPS**$\uparrow$
VOS Methods					
GAM [17]	70.30	85.50	24.1	21.2	24
RVOS [38]	51.90	90.90	61.0	11.9	-
STM [28]	75.60	90.20	22.8	12.9	10
AFB-URR [22]	60.00	84.60	25.5	22.2	9
TVOS [46]	24.00	46.10	58.2	62.1	36
MMA-Net [47]	**83.90**	**91.00**	**11.1**	**10.5**	20
Lane Detection Methods					
LaneNet [41]	72.10	85.80	12.2	20.7	64
SCNN-VGG16 [29]	49.10	90.70	12.8	11.0	25
SAD-ENet [16]	75.50	88.60	17.0	15.2	33
UFLD-R34 [31]	31.00	85.20	11.5	21.5	**124**
LSTR [25]	70.30	88.40	16.3	14.8	40
CLRNet-R18 [49]	57.27	88.99	6.9	13.5	80
CLRNet-R101 [49]	59.41	88.65	**2.1**	12.5	38
ADNet-R18* (Ours)	65.05	94.25	5.0	5.0	-
ADNet-R34* (Ours)	64.97	94.37	4.5	4.9	-
ADNet-R18 (Ours)	89.97	94.23	5.0	5.1	87
ADNet-R34 (Ours)	90.39	**94.38**	4.4	**4.9**	77
ADNet-R101 (Ours)	**90.90**	94.27	4.7	5.0	45

approach also surpasses subsequent anchor-free techniques, such as SGNet [34], by 1.67%. Additionally, our method achieves state-of-the-art performance among segmentation-based and parameter-based methods, but is ranked second after CLRNet [49] on Anchor & Detection-based methods.

TuSimple. The comparison results on TuSimple are presented in Table 3. Due to the limited scenario of TuSimple, the differences between each method are minimal. As can be observed from the table, our method performs better than most of the compared methods.

Table 3: Comparison with state-of-art methods on TuSimple test set.

Methods	**F1@50**$\uparrow$	**Acc**$\uparrow$	**FP**$\downarrow$	**FN**$\downarrow$
SCNN [29]	95.97	96.53	6.17	**1.80**
RESA-R50 [48]	96.93	96.82	3.63	2.48
PolyLaneNet [36]	90.62	93.36	9.42	9.33
E2E-ERFNet [45]	96.25	96.02	3.21	4.28
UFLD-R34 [31]	88.02	95.86	18.91	3.75
UFLDv2-R34 [32]	96.22	95.56	3.18	4.37
SGNet-R34 [34]	-	95.87	-	-
LaneATT-R34 [35]	96.77	95.63	3.53	2.92
CondLaneNet-R101 [24]	97.24	96.54	**2.01**	3.50
FOLOLane-ERFNet [33]	96.59	**96.92**	4.47	2.28
ADNet-R18 (Ours)	96.90	96.23	2.91	3.29
ADNet-R34 (Ours)	**97.31**	96.60	2.83	2.53

4.4. Ablation studies

Overall. We conduct overall ablation study using ResNet18 as the backbone in CULane. The baseline model extracts features from the backbone and FPN, pooling lane features according to the strategy in [13], identical to ADNet, and regressing lane lines using LIoU loss with predefined anchors from [21]. We gradually add SPGU, ALAU, and GLIoU to the baseline, and finally embed LKA with FPN. The overall ablation study results in Table 4 show that GLIoU has the least effect, while SPGU has the greatest effect. The remaining strategies has effects that ranged from big to small, namely ALAU and LKA. Adding SPGU significantly increases the F1@50 score from 72.17 to 76.47, strongly supporting our assumption.

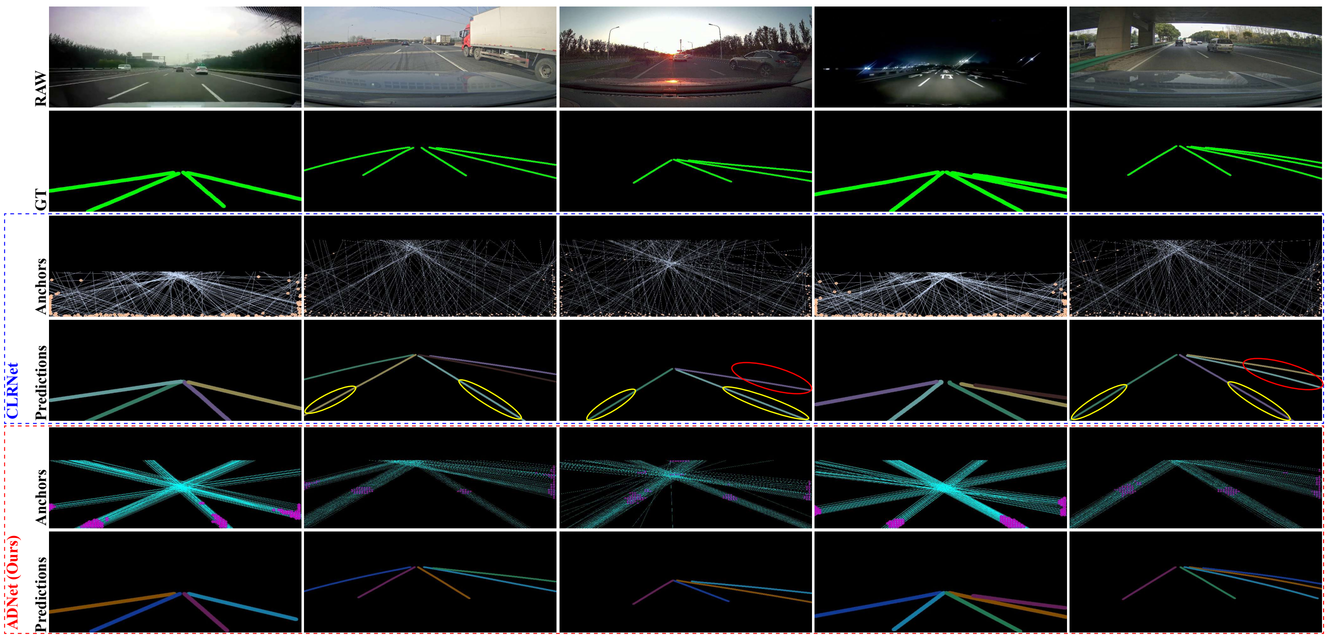


Figure 7: Visualisation results on VIL-100 compare with CLRNet. We visualised every anchors before predictions, yellow oval is applied to highlight the anchor flexibility issue discussed on Section 1. It is evident that our anchors exhibit higher quality compared to CLRNet, which consequently leads to better performance, highlighted by red oval.

Table 4: Overall ablation study of ADNet-R18 on CULane.

Baseline	+SPGU	+ALAU	+GLIoU	+LKA	F1@50
√					72.17
	√				76.47 (+4.30)
	√	√			76.91 (+4.74)
	√	√	√		77.15 (+4.98)
	√	√	√	√	**77.56 (+5.39)**

Effectiveness of guidance map. In ALAU, kernel offsets are obtained from the guidance map as discussed in Section 3.4. This allows us to promote $f_v : \Theta_{xy} \rightarrow \vec{v}$ to $f_v : (\Theta_{xy}, P_{xy}) \rightarrow \vec{v}$, as the starting point and its theta value are spatially related. Our experiments on CULane and VIL-100, as shown in Table 5, confirm this conclusion. Without guidance map represents that we obtain kernel offsets simply from thetas map. The result indicates that when guidance map is added, improvement can be observed on both benchmarks.

Necessity of GLIoU loss. In our overall ablation study, only a 0.24% improvement is brought by GLIoU loss compared to LIoU loss, which is explainable. The scenario we describe in Section 3.3 rarely occurs since on CULane lane lines always ray from three edges of the image. Further experiments on CULane (shown in Table 5) demonstrate that switching the backbone from ResNet18 to ResNet34 with GLIoU loss only brings a 0.18% increment, similar to the phenomenon in Table 4. However, when we conduct the same experiments on VIL-100, both ResNet18 and ResNet34 get a huge boost on F1@50.

Ablation study of LKA. We validate the effectiveness of our LKA by conducting experiments with different backbones and datasets. Results in Table 5 and Table 4 indicate that our LKA not only improves upon plain FPN, but also outperforms the baseline attention module on both VIL-100 and CULane.

Table 5: Ablation study on different components. "w/o" under **Guidance map** represents obtaining kernel offsets from thetas map; "baseline" under **Attention** follows [15].

Dataset	Back-bone	Guidance map	Loss	Attention	F1@50
CULane	R34	w/o	GLIoU	baseline	78.53
	R34	w	GLIoU	baseline	**78.66 (+0.13)**
	R34	w	LIoU	baseline	78.48
	R34	w	GLIoU	baseline	**78.66 (+0.18)**
	R34	w	GLIoU	baseline	78.66
	R34	w	GLIoU	LKA	**78.94 (+0.28)**
VIL-100	R34	w/o	GLIoU	baseline	87.83
	R34	w	GLIoU	baseline	**89.22 (+1.39)**
	R34	w	LIoU	LKA	90.17
	R34	w	GLIoU	LKA	**90.39 (+0.22)**
	R18	w	LIoU	baseline	83.44
	R18	w	GLIoU	baseline	**88.65 (+5.21)**
	R34	w	GLIoU	baseline	89.22
	R34	w	GLIoU	LKA	**90.39 (+1.17)**

5. Conclusion

In this paper, we propose ADNet for lane shape prediction, incorporating SPGU to predict start points and ALAU to aggregate context near lane lines. We introduce GLIoU loss to address limitations of LIoU loss and modify the small kernel attention module into LKA. Our algorithm outperforms current state-of-the-art methods on VIL-100 and achieves nearest state-of-the-art on CULane and TuSimple.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under No.62276061 and 62006041.

References

[1] Hala Abualsaud, Sean Liu, David B Lu, Kenny Situ, Akshay Rangesh, and Mohan M Trivedi. Laneaf: Robust multi-lane detection with affinity fields. *IEEE Robotics and Automation Letters*, 6(4):7477–7484, 2021. 2, 7

[2] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane markers using maps. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 6

[3] Gong Cheng, Liming Cai, Chunbo Lang, Xiwen Yao, Jinyong Chen, Lei Guo, and Junwei Han. Spnet: Siamese-prototype network for few-shot remote sensing image scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2021. 3

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3, 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[6] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11963–11975, 2022. 4

[7] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 4

[8] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 3

[9] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022. 7

[10] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 5

[11] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *arXiv preprint arXiv:2209.08575*, 2022. 4

[12] Jianhua Han, Xiajun Deng, Xinyue Cai, Zhen Yang, Hang Xu, Chunjing Xu, and Xiaodan Liang. Laneformer: Object-aware row-column transformers for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 799–807, 2022. 7

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[14] 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. 6, 11

[15] 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. 4, 8, 11

[16] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1013–1021, 2019. 7

[17] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8953–8962, 2019. 7

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] 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*, 23(7):8949–8958, 2021. 3, 7

[20] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 3

[21] 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, 2, 3, 7, 11

[22] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. *Advances in Neural Information Processing Systems*, 33:3430–3441, 2020. 7

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4, 6

[24] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: a top-to-down lane detection framework based on conditional convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3773–3782, 2021. 1, 2, 7

[25] 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. 7

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[28] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 7

[29] 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, 2, 6, 7, 11

[30] Jonah Philion. Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11582–11591, 2019. 7

[31] 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, 2, 7

[32] Zequn Qin, Pengyi Zhang, and Xi Li. Ultra fast deep lane detection with hybrid anchor driven ordinal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 7

[33] 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. 3, 7

[34] Jinming Su, Chao Chen, Ke Zhang, Junfeng Luo, Xiaoming Wei, and Xiaolin Wei. Structure guided lane detection. *arXiv preprint arXiv:2105.05403*, 2021. 7

[35] 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, 6, 7

[36] 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. 7

[37] TuSimple. Tusimple lane detection benchmark(2017). `https://github.com/TuSimple/tusimple-benchmark`. 1, 2, 6

[38] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5277–5286, 2019. 7

[39] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019. 3

[40] Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, and Tianzhu Zhang. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1392–1401, 2022. 1, 3, 4, 5, 7

[41] Ze Wang, Weiqiang Ren, and Qiang Qiu. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv preprint arXiv:1807.01726*, 2018. 7

[42] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 689–704. Springer, 2020. 7

[43] Jiaxing Yang, Lihe Zhang, and Huchuan Lu. Lane detection with versatile atrousformer and local semantic guidance. *Pattern Recognition*, 133:109053, 2023. 7

[44] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9657–9666, 2019. 3

[45] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-end lane marker detection via row-wise classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1006–1007, 2020. 7

[46] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2020. 7

[47] 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. 2, 6, 7, 11

[48] 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. 2, 7

[49] 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, 2, 3, 4, 5, 7, 11, 13

[50] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 850–859, 2019. 3

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 [21]. 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:

$$x_i = s_x \cdot w + \frac{(y_i - s_y)h}{tan(1-\theta)\cdot\pi}\ ,\ y_i \in Y, \qquad (18)$$

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 6: Effectiveness of different MSA designs. Models are trained and tested on CULane [29] 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 [15]) to our final LKA. Apart from the baseline, Figure 9 presents our three candidate designs of the MSA module on LKA. MSA-A (Figure 9(b)) replaces the $DConv$ on the baseline with a multi-channel strip $DConv$, MSA-B (Figure 9(c)) further adds an identical forward path, and MSA-C (Figure 9(d)) replaces Liner with $DConv$ with a kernel size of 5×5. A series of experiments are shown in Table 6. Our final mechanism, MSA-C, exhibits superior performance to the other designs.

C. GLIoU

To emphasise the effectiveness of GLIoU, we present the results on the validation set of VIL-100 [47] in chart form.

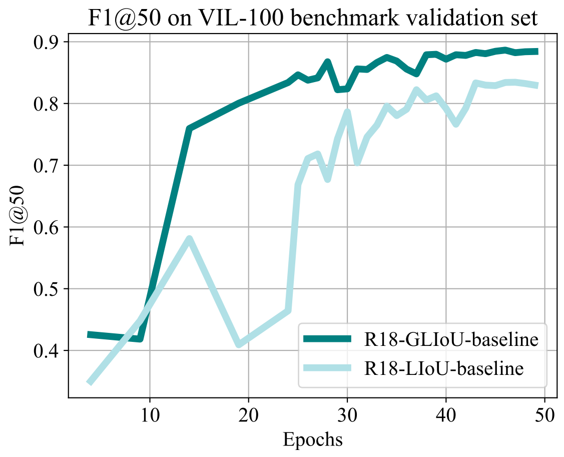


Figure 8: Illustration of the effectiveness of GLIoU on VIL-100.

The validation results are presented in Figure 8. "R18-GLIoU-baseline" represents ResNet18 [14] as the backbone, GLIoU as the regression loss, and "baseline" [15] as the attention module.

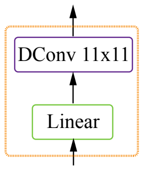
The results in Figure 8 demonstrate that using GLIoU as the regression loss function generally yields better performance than LIoU [49] across all documented epochs and reaches saturation more quickly than LIoU.

Table 7: Effectiveness of anchors number. Models are trained and tested on CULane and VIL-100 using different backbones. "R18" stands for ResNet18.

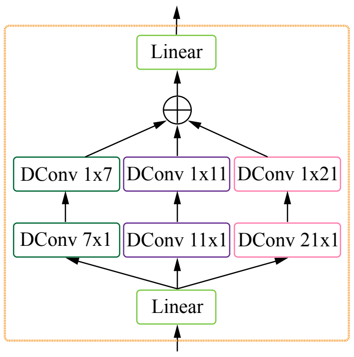
Dataset	Backbone	Anchors	F1@50(%)	Precision(%)	Acc(%)
CULane	R18	100	77.53	85.56	-
		200	77.51	85.20	-
		300	77.56	85.01	-
		400	77.63	84.85	-
		500	77.61	86.62	-
VIL-100	R18	50	90.02	90.50	94.10
		75	90.04	90.29	94.21
		100	89.97	90.09	94.23
	R101	50	90.83	91.10	94.14
		75	90.90	90.99	94.27
		100	90.90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

D. Supervision for heat map

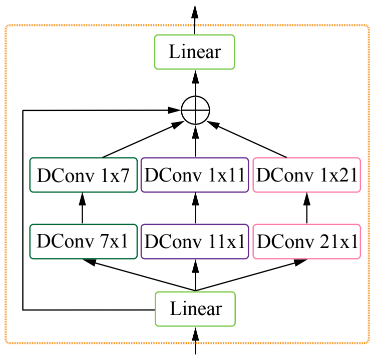
In our framework, we propose supervision for the start points heat map using a non-normalised Gaussian kernel. The hyperparameter σ controls the size of the region that potentially contains start points. Scenarios that typically consist of invisible start points tend to favour higher σ values compared to scenarios with clear and eye-catching start points. Figure 10 visualises the start points heat map supervision using four different σ values on CULane. On the



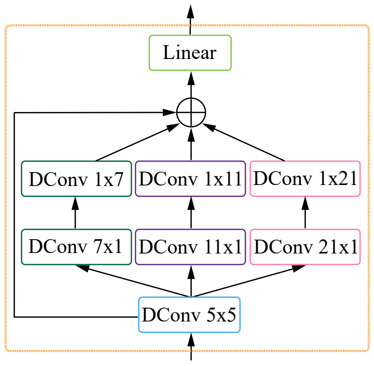
(a) baseline



(b) MSA-A

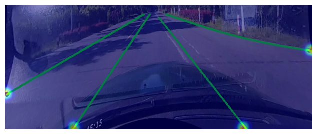


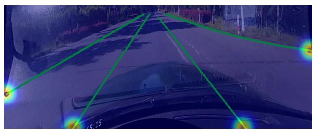
(c) MSA-B

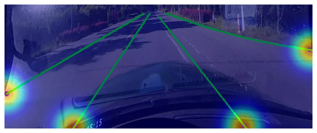


(d) MSA-C

Figure 9: Illustration of different designs of MSA.


(a) $\sigma = 1$


(b) $\sigma = 2$


(c) $\sigma = 4$

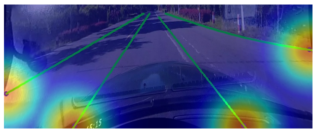

(d) $\sigma = 8$

Figure 10: Illustration of start points heat map supervision with different σ on CULane.

activation map, red regions potentially contain start points, while blue regions have a lower likelihood.

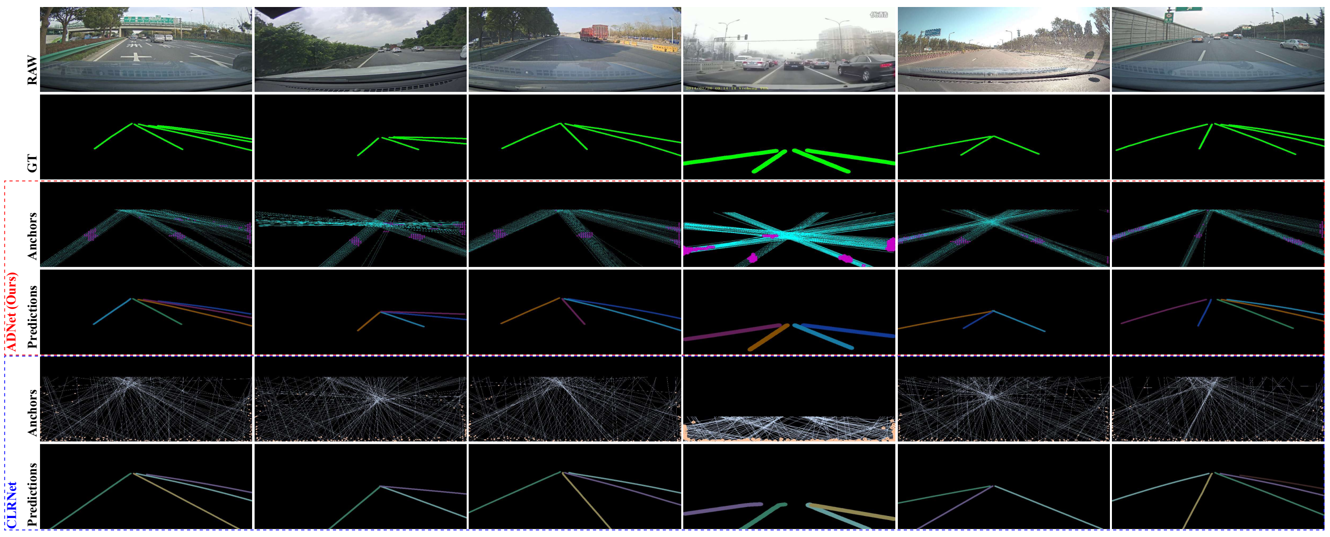
Table 8 shows the results of our experiments conducted on CULane and VIL-100. For CULane, we choose σ values of 2, 4, and 8 since start points on CULane are usually invisible. For VIL-100, we choose σ values of 1, 2, and 4 since start points are generally clear and featured in this dataset. It can be 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 8: 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.

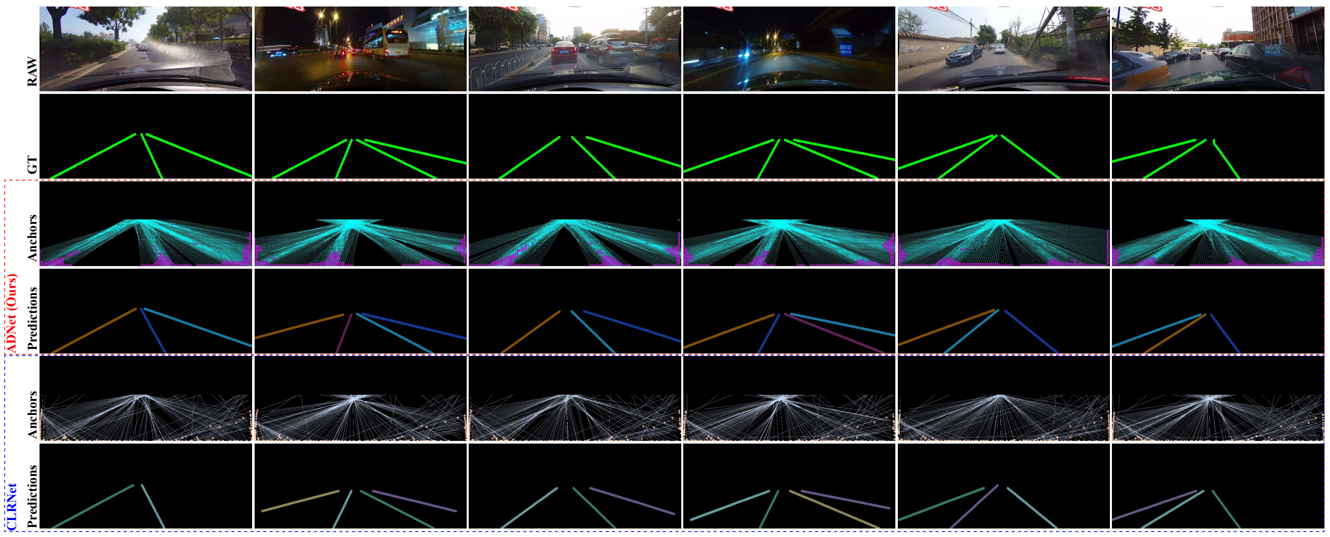
Dataset	σ	F1@50(%)	Precision(%)	Acc(%)
CULane	2	76.87	83.60	-
	4	**77.56**	85.01	-
	8	77.51	**86.25**	-
VIL-100	1	89.69	**90.24**	94.14
	2	**89.97**	90.09	**94.23**
	4	86.66	85.94	94.04

E. Number of anchors

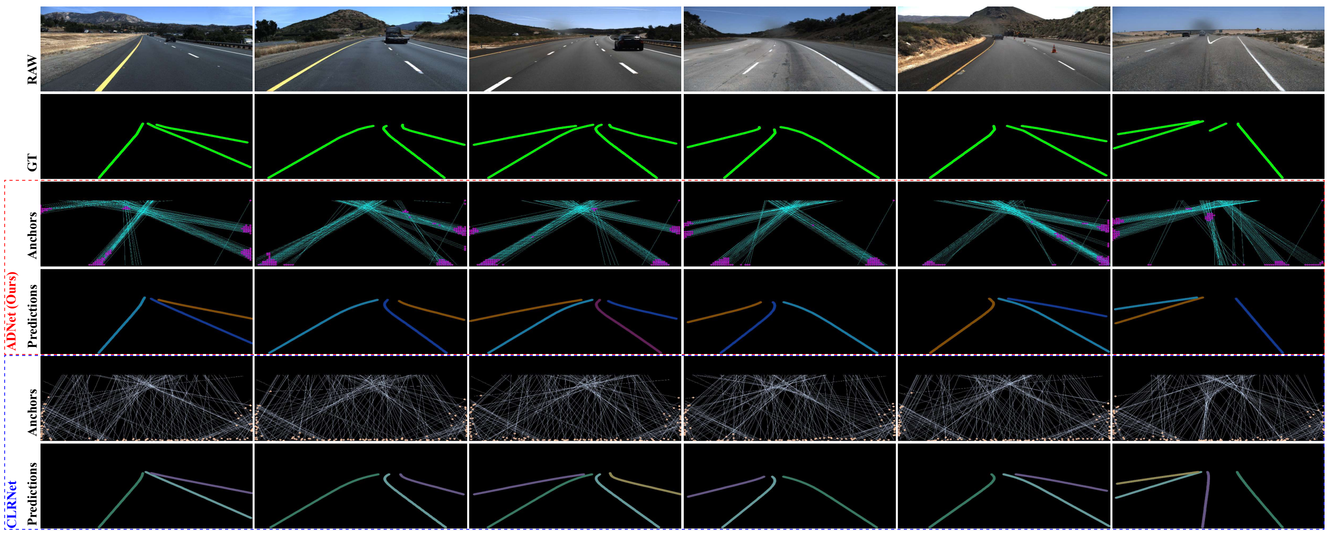
The number of anchors can be regarded as a hyperparameter during the training and validation phases. Our experiments are conducted on CULane and VIL-100, using different backbones, ResNet18 and ResNet101. The training phase follows the procedure outlined in our paper. Results are presented in Table 7. 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) Visualisation results on VIL-100



(b) Visualisation results on CULane



(c) Visualisation results on TuSimple

Figure 11: More visualisation results on VIL-100, CULane and TuSimple compared with CLRNet [49]. Best view in zoom.