A Keypoint-based Global Association Network for Lane Detection

Jinsheng Wang[1,3*] Yinchao Ma[2,3*] Shaofei Huang[3*] Tianrui Hui[4,5]
Fei Wang[2] Chen Qian[3] Tianzhu Zhang[2†]
[1] Peking University [2] University of Science and Technology of China [3] SenseTime Research
[4] Institute of Information Engineering, Chinese Academy of Sciences
[5] School of Cyber Security, University of Chinese Academy of Sciences

Abstract

Lane detection is a challenging task that requires predicting complex topology shapes of lane lines and distinguishing different types of lanes simultaneously. Earlier works follow a top-down roadmap to regress predefined anchors into various shapes of lane lines, which lacks enough flexibility to fit complex shapes of lanes due to the fixed anchor shapes. Lately, some works propose to formulate lane detection as a keypoint estimation problem to describe the shapes of lane lines more flexibly and gradually group adjacent keypoints belonging to the same lane line in a point-by-point manner, which is inefficient and time-consuming during postprocessing. In this paper, we propose a Global Association Network (GANet) to formulate the lane detection problem from a new perspective, where each keypoint is directly regressed to the starting point of the lane line instead of point-by-point extension. Concretely, the association of keypoints to their belonged lane line is conducted by predicting their offsets to the corresponding starting points of lanes globally without dependence on each other, which could be done in parallel to greatly improve efficiency. In addition, we further propose a Lane-aware Feature Aggregator (LFA), which adaptively captures the local correlations between adjacent keypoints to supplement local information to the global association. Extensive experiments on two popular lane detection benchmarks show that our method outperforms previous methods with F1 score of 79.63% on CULane and 97.71% on Tusimple dataset with high FPS.

1. Introduction

Autonomous driving [10] has drawn remarkable attention of researchers from both academia and industry. In order to ensure the safety of the car during driving, the autonomous system needs to keep the car moving along the lane lines on the road, requiring accurate perception of the lane lines. Thus, lane detection plays an important role in the autonomous driving system, especially in Advanced Driver Assistance System (ADAS).

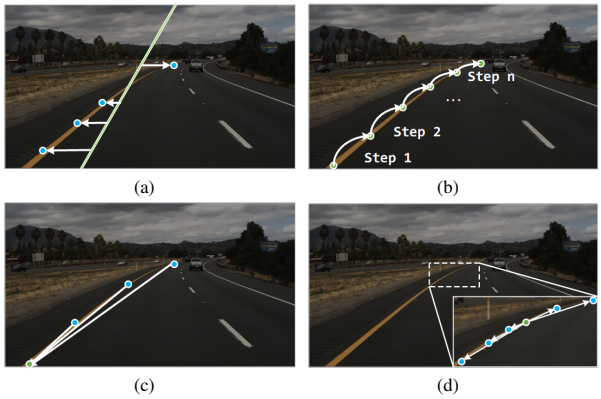

Figure 1. (a) Anchor-based methods, which regress the predefined anchors into the shape of lanes. (b) Keypoint-based methods, which predict offsets between keypoint to its neighbourhood to group them one-by-one. (c) Illustration of our GANet, which directly regresses each keypoint to its belonged lane by predicting offset between each keypoint and the starting point of its corresponding lane line. (d) Illustration of our LFA module, which correlates each keypoint with its adjacent points for local information supplement.

Given a front-viewed image taken by a camera mounted on the vehicle, lane detection aims to produce the accurate shape of each lane line on the road. Due to the slender shapes of lane lines and the need for instance-level discrimination, it is crucial to formulate lane detection task appropriately. Inspired by the anchor-based object detection methods [22], some works [10, 25] follow a top-down design as illustrated in Figure 1a. Similar to object detection, a group of straight lines with various orientations are defined as anchors. Points on anchors are regressed to lane lines by predicting the offsets between anchor points and lane

*Equal contribution
†Corresponding author (tzzhang@ustc.edu.cn)

points. Afterward, Non-Maximum Suppression (NMS) is applied to select lane lines with the highest confidence. Although this kind of method is efficient in lane discrimination, it is inflexible because of the predefined anchor shapes. The strong shape prior limits the ability of describing various lane shapes, resulting in sub-optimal performances of these methods.

To describe complex shapes of lane lines flexibly, Qu *et al*. [21] propose to formulate lane detection as a keypoint estimation and association problem, which takes a bottom-up design as illustrated in Figure 1b. Concretely, lanes are represented with a group of ordered key points evenly sampled in a sparse manner. Each key point is associated with its neighbours by estimating the spatial offsets between them. In this way, key points belonging to the same lane are integrated into a continuous curve iteratively. Though keypoint-based methods are flexible on the shape of lane lines, it is inefficient and time-consuming to associate only one keypoint to its belonged lane line at each step. Besides, the point-by-point extension of keypoints is easy to cause error accumulation due to the lack of global view. Once a particular keypoint is wrongly associated, estimation of the rest part of the lane line will fail.

To overcome the above limitations, we formulate the lane detection problem from a new keypoint-based perspective where each keypoint is directly regressed to its belonged lane, based on which a novel pipeline named Global Association Network (GANet) is proposed. As illustrated in Figure 1c, each lane line is represented uniquely with its starting point, which is easy to determine without ambiguity. To associate a keypoint properly, we estimate the offset from the keypoint to its corresponding starting point. Keypoints whose approximated starting points fall into the same neighborhood area will be assigned to the same lane line instance, thus separating keypoints into different groups. Different from previous keypoint-based method [21], our assignment of keypoints to their belonged lanes is independent of each other and makes the parallel implementation feasible, which greatly improves the efficiency of post-processing. Besides, the keypoint association is more robust to the accumulated single-point errors since each keypoint owns a global view.

Although keypoints belonging to the same lane line are integrated during post-processing, it is important to ensure the correlations between adjacent points in order to obtain a continuous curve. To this end, we develop a local information aggregation module named Lane-aware Feature Aggregator (LFA) to enhance the correlations between adjacent keypoints. To adapt to the slender and long shapes of lanes, we modify the sampling positions of the standard 2D deformable convolution [3] by predicting offsets to adjacent points to sample within a local area on the lane each time. In this way, features of each keypoint are aggregated with other adjacent points, thus acquiring more representative features. We further add an auxiliary loss to facilitate estimating the offset predicted on each key point. Our LFA module complements the global association process to enable both local and global views, which is essential for dense labeling tasks like lane detection.

Our contributions are summarized as follows:

- We propose a novel Global Association Network (GANet) to formulate lane detection from a new keypoint-based perspective which directly regress each keypoint to its belonged lane. To the best of our knowledge, we are the first to regress keypoints in a global manner, which is more efficient than local regression.
- We develop a local information aggregation module named as Lane-aware Feature Aggregator (LFA) to enhance correlations among adjacent keypoints to supplement local information.
- Our proposed GANet achieves state-of-the-art performances on two popular benchmarks of lane detection with faster speed, which shows a superior performance-efficiency trade-off and great potential of our global association formulation.

2. Related Works

2.1. Lane Detection Methods

Lane detection aims at obtaining the accurate shape of lane lines as well as distinguishing between them. According to the way of lane modeling, current deep-learning-based methods can be roughly divided into several categories. We will elaborate on these methods separately in this section.

Segmentation-based methods. Segmentation-based methods model lane line detection as a per-pixel classification problem, with each pixel classified as either lane area or background [6, 8, 16, 18]. To distinguish different lane lines, SCNN [18] treats different lane lines as different categories and thus lane detection is transformed into a multi-class segmentation task. A slice-by-slice CNN structure is also proposed to enable message passing across rows and columns. In order to meet the real-time requirement in practice, ENet-SAD [6] applies a self-attention distillation mechanism for contextual aggregation so as to allow the use of a lightweight backbone. LaneNet [16] adopts a different way of lane representation by casting lane detection as an instance segmentation problem. A binary segmentation branch and an embedding branch are included to disentangle the segmented results into lane instances. Different from LaneNet, our method use offsets instead of embedding features to cluster each lane lines, which is more efficient and time-saving.

Detection-based methods. This kind of method usually follows a top-down manner to predict lane lines. Among them, anchor-based methods [10, 25, 28] design line-like anchors and regress the offsets between sampled points and predefined anchor points. Non-Maximum Suppression (NMS) is then applied to select lane lines with the highest confidence. LineCNN [10] uses straight rays emitted from the image boundaries with certain orientations as a group of anchors. Curve-NAS [28] defines anchors as vertical lines and further adopts neural architecture search (NAS) to search for better backbones. LaneATT [25] proposes an anchor-based pooling method and an attention mechanism to aggregate more global information. Another kind of methods [14, 20] formulates lane detection as a row-wise classification problem. For each row, the model predicts the locations that possibly contain lane lines.

Keypoint-based methods. Inspired by human pose estimation, some works treat lane detection as a keypoint estimation and association problem. PINet [9] uses a stacked hourglass network [17] to predict keypoint positions and feature embedding. Different lane instances are clustered based on the similarity between feature embeddings. FOLOLane [21] produces a pixel-wise heatmap with the same resolution as input to obtain points on lanes. A local geometry construction manner is also developed to associate keypoints belonging to the same lane instance. Our GANet adopts a more efficient postprocessing approach, which needs neither feature embeddings nor local association to cluster or reconstruct the whole lane. Each keypoint finds its corresponding lane by adding its coordinate with the offset to the lane line start points in a parallel manner.

2.2. Deformable Modeling

Traditional CNNs are inherently limited to model irregular structures due to the fixed grid-like sampling ranges of convolution operations. To overcome this limitation, Dai *et al*. [3] proposes deformable convolution to adaptively aggregate information within local areas. Compared with standard convolutions, 2D offsets obtained by an extra convolution are added at each spatial location during sampling to enable free-form deformation of the sampling grid. Through the learned offsets, the receptive field and the sampling location of convolutions are adaptively adjusted according to the random scale and shape of objects. The spirit of deformable modeling has been applied in many tasks such as object detection [30, 34], object tracking [33] and video comprehension [2, 29, 31]. RepPoints [30] models an object as a set of points and predicts the offsets of these points to the object center with deformable convolutions. This deformable object representation provides accurate geometric localization for object detection as well as adaptive semantic feature extraction. Ying *et al*. [31] proposes deformable 3D convolution to explore spatio-temporal information and realize adaptive motion comprehension for video super-resolution. Different from these methods, our LFA module adapts to the long structure of lane lines and restricts the range of feature aggregation to adjacent points on each lane with lane-aware deformable convolutions.

3. Method

The overall architecture of our proposed Global Association Network (GANet) is illustrated in Figure 2. Given a front-viewed image as input, a CNN backbone together with an FPN [12] neck is adopted to extract multi-level visual representations of the input images. For better feature learning, a self-attention layer [27] is further inserted between the backbone and the neck to obtain rich context information. In the decoder, a keypoint head and an offset head are exploited to generate confidence map and offset map respectively. Both heads are composed of fully convolutional layers. We further devise a Lane-aware Feature Aggregator module before keypoint head to enhance the local correlations between adjacent keypoints, which facilitates to produce continuous lane lines. For each lane instance, we first obtain its starting point as cluster centroid by selecting points with value less than 1 over the offset map. Afterward, keypoints belonging to the same lane are clustered around the sampled starting point with the combination of the confidence map and offset map to construct the complete lane line.

3.1. Global Keypoint Association

3.1.1 Keypoint Estimation

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the goal of our GANet is to predict a collection of lanes $L = \{l_1, l_2, ..., l_N\}$, where N is the total number of lanes, with each lane line being denoted with K sampled keypoints as:

$$l_i = \{p_i^1, p_i^2, ..., p_i^K\}_{i=1}^N, \tag{1}$$

where $p_i^j = (x_i^j, y_i^j)$ denotes the coordinate of the j-th keypoint on the i-th lane. To estimate all the keypoints, we develop a keypoint head to produce a confidence map $\hat{Y} \in \mathbb{R}^{\frac{H}{r}\times\frac{W}{r}}$, where r is the output stride. The confidence map represents the probability of each location being a keypoint on the lane. As shown in Figure 2(a), the brighter location indicates a higher probability.

During the training phase, we sample K keypoints on each lane line as ground truth keypoints and then splat them all onto a confidence map $Y \in \mathbb{R}^{\frac{H}{r}\times\frac{W}{r}}$ using a non-normalized Gaussian kernel $Y_{yx} = exp(-\frac{(x-\tilde{x})^2+(y-\tilde{y})^2}{2\sigma^2})$, where $\tilde{x}$ and $\tilde{y}$ denote the coordinate of each keypoint and the standard deviation σ depends on the scale of input. If there is overlap between two Gaussian maps, we take the element-wise maximum between them.

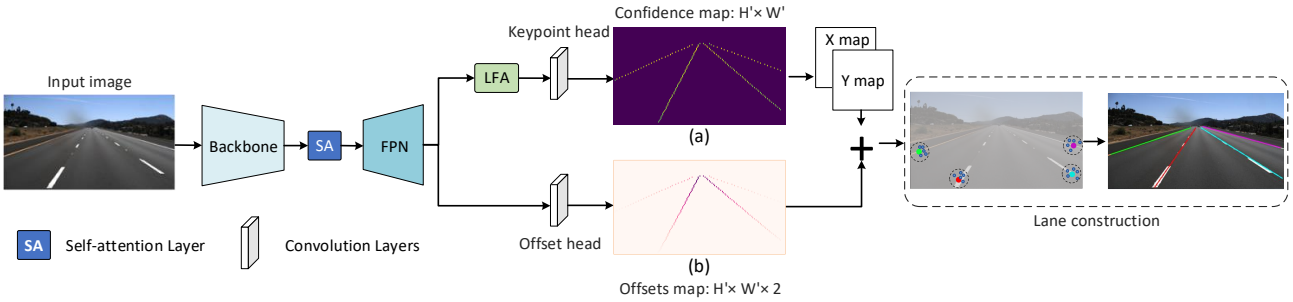

Figure 2. The overall architecture of GANet. Given a front-viewed image as input, a CNN backbone followed by a Self-Attention layer (SA) and an FPN neck are used to extract multi-scale visual features. In the decoder, a keypoint head and an offset head are used to generate confidence map and offset map respectively, which are then combined to cluster keypoints into several groups, with each group indicating a lane line instance. Our LFA module is applied before the keypoint head to better capture local context over lane lines for keypoint estimation.

We adopt penalty-reduced focal loss [13] to deal with the imbalance between keypoint regions and non-keypoint regions as follows:

$$\mathcal{L}_{point} = \frac{-1}{H' \times W'} \sum_{yx} \begin{cases} (1-\hat{Y}_{yx})^{\alpha} log(\hat{Y}_{yx})) & Y_{yx}=1 \\ (1-Y_{yx})^{\beta} \hat{Y}_{yx}^{\alpha} log(1-\hat{Y}_{yx})) & otherwise, \end{cases} \tag{2}$$

where α and β are hyper-parameters of focal loss and $H' \times W'$ denotes $\frac{H}{r} \times \frac{W}{r}$. The subscript yx represents obtaining the value at coordinate (x, y).

Due to the output stride r, the point (x_i^j, y_i^j) of the input image is mapped to the location $(\lfloor \frac{x_i^j}{r} \rfloor, \lfloor \frac{y_i^j}{r} \rfloor)$, which can cause performance degradation. To address this quantization error, we additionally predict a compensation map $\hat{\delta}_{yx}$ and apply L1 loss to keypoint locations only:

$$\mathcal{L}_{quant} = \frac{1}{H' \times W'} \sum_{yx} \left| \hat{\delta}_{yx} - \delta_{yx} \right|, \tag{3}$$

where $\delta_{yx} = (\frac{x_i^j}{r} - \lfloor \frac{x_i^j}{r} \rfloor, \frac{y_i^j}{r} - \lfloor \frac{y_i^j}{r} \rfloor)$ denotes the ground truth of quantization compensation map. This part is not shown in Figure 2 for simplicity.

3.1.2 Starting Point Regression

To distinguish different lane lines, we propose to use the starting point to represent each lane instance uniquely due to its stability and largest margins between each other. Instead of regressing the absolute coordinate (sx_i, sy_i) of the starting point directly, we regress the offset from each keypoint to it, which can be defined as:

$$(\Delta x_i^j, \Delta y_i^j) = (sx_i, sy_i) - (x_i^j, y_i^j), \tag{4}$$

Thus, we can generate the ground truth offset map O_{yx} with the shape of $\frac{H}{r} \times \frac{W}{r} \times C$. In particular, the subscript yx denotes the value on location (x_i^j, y_i^j) which is equal to $(\Delta x_i^j, \Delta y_i^j)$ while other locations have zero values. $C = 2$ contains the x-direction and y-direction offsets respectively.

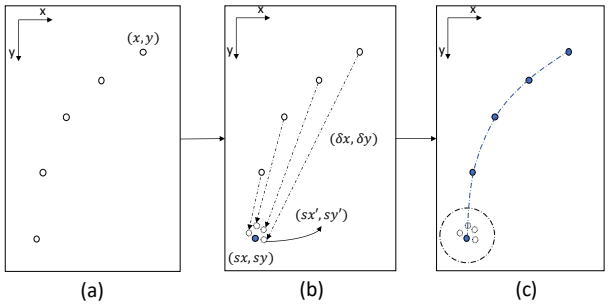

Figure 3. Illustration of lane construction. (a) Valid keypoints are selected from the confidence map. (x, y) is taken as an example. (b) Starting point (sx, sy) (blue dot) is sampled first. The rest keypoints point to the starting point with the predicted offset $(\delta x, \delta y)$ and estimate the coordinate of the starting points as $(sx', sy') = (x, y) + (\delta x, \delta y)$ (hollow dots). (c) Keypoints that point to the neighbourhood of starting point (sx, sy) are grouped as a whole lane.

In order to estimate the offset map $\hat{O}_{yx}$, we introduce an offset head as shown in Figure 2. Similarly, L1 loss is used to constrain the offset map as follows:

$$\mathcal{L}_{offset} = \frac{1}{H' \times W'} \sum_{yx} \left| \hat{O}_{yx} - O_{yx} \right|, \tag{5}$$

The supervision applies only on keypoint locations and the rest locations are ignored.

3.1.3 Lane Construction

The pipeline of lane construction is presented in Figure 3, which includes obtaining the locations of all the possible

lane points and then grouping them into different lane instances. We first apply a 1×3 max pooling layer on the keypoint confidence map $\hat{Y}$ to select points of maximum responses within a horizontal local region as valid keypoints, which is shown in Figure 3(a). Then, we group them to describe each lane as an ordered list of keypoints as follows:

$$l = \{(sx, sy), (x^2, y^2), (x^3, y^3), ..., (x^K, y^K)\}, \quad (6)$$

where (sx, sy) denotes the starting point of the lane and $(x^j, y^j), j \in [2, K]$ are the subsequent keypoints.

To obtain the starting point of each lane, we select keypoints whose values are less than 1 on the offset map as the candidate starting points. Since there might be multiple keypoints matching the above criteria within the same local region, the geometric center point of the region is chosen to ensure uniqueness. By this means, instances of all the lanes are preliminarily determined with their starting points.

Afterward, we associate the rest keypoints to their belonged lanes according to the estimated offsets between keypoints and corresponding starting points, which is shown in Figure 3(b). Each keypoint estimates the coordinate of the lane line starting point as follows:

$$(sx', sy') = (x, y) + (\delta x, \delta y), \quad (7)$$

where (x, y) is the coordinate of the observed keypoint and $(\delta x, \delta y) = O_{yx}$ denotes the corresponding offset obtained in Section 3.1.2. The keypoint (x, y) is associated to the i-th lane only if the distance between (sx', sy') and (sx, sy) is less than a predefined threshold θ_{dis}. As shown in Figure 3(c), keypoints that point to the neighborhood of the same starting point are grouped to produce a whole lane. The above procedures are done by matrix operations to ensure parallel keypoints association.

3.2. Lane-aware Feature Aggregator

Traditional 2D convolutions sample features within a fixed grid-like region, which is not suitable for handling the slender shapes of lane lines. Inspired by Dai *et al.* [3], we propose a Lane-aware Feature Aggregator (LFA) module to adaptively gather information from adjacent points on the lanes, so as to enhance the local feature representation of each keypoint. The illustration of our LFA module is shown in Figure 4. Take a specific keypoint as an example, we first use a convolution layer to predict the offset between it and its surrounded M keypoints on the same lane as follows:

$$\Delta P_i = \phi(\mathcal{F}(p_i)), \quad (8)$$

where p_i denotes the coordinate of the i-th keypoint, $\mathcal{F}(p_i)$ denotes the feature representation of the i-th keypoint and $\Delta P_i = \{\Delta p_i^m | m = 1, ..., M\} \in \mathbb{R}^{2M}$ denotes the predicted offsets. Afterwards, features of adjacent points are integrated with a deformable convolution to aggregate context of the i-th keypoint as:

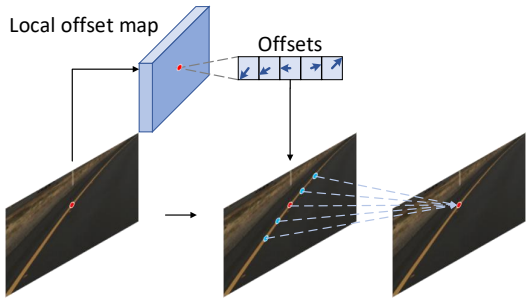

Figure 4. Illustration of LFA module. The red dot denotes the observed keypoint. We first predict offsets between the red dot and its adjacent keypoints (in blue) and then gather features of these keypoints to enhance the context of the red keypoint.

$$\hat{\mathcal{F}}(p_i) = \sum_{m=1}^{M} w_m \cdot \mathcal{F}(p_i + \Delta p_i^m), \quad (9)$$

where $w_m, m = 1, ..., M$ is the weight of the convolution and $(\cdot)$ means multiplication.

To enhance the ability of LFA for learning the local shapes of lane lines, we further introduce an auxiliary loss to supervise the offsets ΔP_i. We denote the ground truth of offsets between the i-th keypoint and the keypoints on the corresponding lane line as $\Delta G_i = \{\Delta g_i^k | k = 1, ..., K\}$, which is calculated with $\Delta g_i^k = g_i^k - p_i$, where g_i^k is the ground-truth coordinate of the k-th keypoint on the same lane line with the i-th keypoint.

As is shown in Figure 5, a matching need to be established between Δp_i and Δg_i. We search for an assignment σ with the lowest matching cost:

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{m}^{M} \mathcal{L}_{match}(\Delta p_i^m, \Delta {g_i}^{\sigma(m)}), \quad (10)$$

where $\mathcal{L}_{match} = L_2(\Delta p_i^m, \Delta {g_i}^{\sigma(m)})$. Following prior works [1, 23], the Hungarian algorithm is adopted to efficiently compute the optimal assignment. SmoothL1 loss is then applied to supervise the prediction of adjacent keypoints:

$$\mathcal{L}_{aux} = \frac{1}{KNM} \sum_{i=1}^{KN} \sum_{m=1}^{M} SmoothL1(\Delta p_i^m, \Delta {g_i}^{\hat{\sigma}(m)}), \quad (11)$$

where K denotes the number of keypoints on each lane line, N denotes the number of lane lines and M denotes the number of sampled adjacent keypoints.

The total loss function is the combination of different

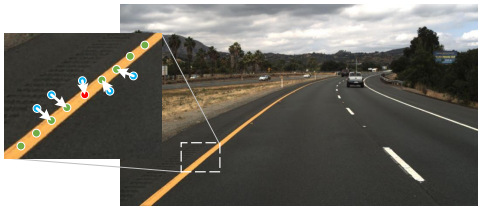

Figure 5. Illustration of the matching between predict points and their ground truth. The red dot is the observed keypoint. The blue dots are the predicted locations of adjacent keypoints. The green dots are the ground-truth locations of adjacent keypoints on the lane line.

losses with corresponding coefficients:

$$\mathcal{L}_{total} = \lambda_{point}\mathcal{L}_{point} + \lambda_{quant}\mathcal{L}_{quant} + \lambda_{offset}\mathcal{L}_{offset} + \lambda_{aux}\mathcal{L}_{aux}. \quad (12)$$

4. Experiments

In this section, we first introduce the experimental setting of our method. The next subsection discusses the results for each dataset. Ablation experiments for each module is presented in the last subsection.

4.1. Experimental Setting

4.1.1 Datasets and Evaluation Metrics

We conduct experiments on two popular lane detection benchmarks including CULane [18] and TuSimple [26].

CULane: CULane dataset contains $88,880$ training images and $34,680$ testing images, including both urban and highway scenes. The test images are classified as 9 different scenarios. F1 measure is the only metric for evaluation, which is based on IoU. A predicted lane whose IoU is greater than 0.5 is judged as true positive (TP), otherwise false positive (FP) or false negative (FN). F1 measure is defined as the harmonic average of precision and recall.

TuSimple: TuSimple is a real highway dataset which consists of $3,626$ images for training and $2,782$ images for testing. The main evaluation metric of TuSimple dataset is accuracy, which is formulated as follows:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}$$

where C_{clip} is the number of points correctly predicted by the model and S_{clip} is the total number of points in the clip (or image). A predicted point is considered correct only if it is within 20 pixels to the ground truth point. The predicted lane with accuracy greater than 85% is considered as a true positive. We also report the F1 score in the following experiments.

Model version	Backbone	FPN layers	Output downscale
GANet-S	resnet-18	3	8
GANet-M	resnet-34	3	8
GANet-L	resnet-101	4	4

Table 1. Details of different versions of GANet.

4.1.2 Implementation Details

We choose ResNet-18, ResNet-34 and ResNet-101 [5] as the backbones to form three different versions of GANet, which are referred as GANet-S, GANet-M and GANet-L. The detail of each version is shown in Table 1. We first resize the input images to 800×320 during the training and testing phases. The number of sampled points in LFA is set as $M = 9$. The loss weights are set as $\lambda_{point} = 1.0$, $\lambda_{quant} = 1.0$, $\lambda_{offset} = 0.5$, $\lambda_{aux} = 1.0$. The hyper-parameters α and β in Equation 2 are set as 2 and 4 respectively. For optimization, we used Adam optimizer and poly learning rate decay with an initial learning rate of 0.001. We train 300 and 40 epochs for Tusimple and CULane respectively with a batchsize of 32 per GPU. Data augmentation is applied to the training phase, including random scaling, cropping, horizontal flipping, random rotation, and color jittering. In the test phase, we set the threshold of keypoints as 0.4 and θ_{dis} for keypoint association as 4. Training and testing are both performed on Tesla-V100 GPUs.

4.2. Quantitative Results

4.2.1 Results on CULane

The results on CULane test set are shown in Table 2. Our GANet-L achieves the state-of-the-art result on CULane dataset with 79.63% F1 score and 63 FPS, which exceeds models of similar size, like LaneATT-ResNet122, with large margins on both performance and speed. Compared with another keypoint-based method, FOLOLane-ERF [21], our GANet-S achieves a comparable performance of 78.79% F1 score but runs 3.8 times faster, which shows a superior trade-off between performance and efficiency and demonstrate the speed advantage of our global association formulation. Furthermore, our methods achieve the highest F1 score in six scenarios, especially in Curve scenario. Our GANet-L achieves 77.37% in this scenario and outperforms previous state-of-the-art method, ERF-E2E [32], with more than 5%, indicating the superiority of our method in describing complex lane line shapes.

4.2.2 Results on TuSimple

The comparison results on TuSimple test set are shown in Table 4. Our GANet-S outperforms all other methods and achieves the highest F1 score of 97.71% with high FPS. It is worth noting GANet-S exceeds UFast-ResNet34 and

Method	Total	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Curve	Cross	Night	FPS
Segmentation-based											
SCNN [18]	71.60	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10	7.5
ENet-SAD [7]	70.80	90.10	68.80	60.20	65.90	41.60	84.00	65.70	1998	66.00	75
Detection-based											
FastDraw [19]	-	85.90	63.60	57.00	69.90	40.60	79.40	65.20	7013	57.80	90.3
UFAST-ResNet18 [20]	68.40	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10	**322.5**
UFAST-ResNet34 [20]	72.30	90.07	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70	175
ERF-E2E [32]	74.00	91.00	73.10	64.50	74.10	46.60	85.80	71.90	2022	67.90	-
CurveLanes-NAS-L [28]	74.80	90.70	72.30	67.70	70.10	49.40	85.80	68.40	1746	68.90	-
LaneATT-ResNet18 [25]	75.13	91.17	72.71	65.82	68.03	49.13	87.82	63.75	**1020**	68.58	250
LaneATT-ResNet34 [25]	76.68	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72	171
LaneATT-ResNet122 [25]	77.02	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81	26
Keypoint-based											
FOLOLane-ERF [21]	78.80	92.70	77.80	**75.20**	79.30	52.10	89.00	69.40	1569	**74.50**	40
GANet-S	78.79	93.24	77.16	71.24	77.88	**53.59**	89.62	75.92	1240	72.75	153
GANet-M	79.39	**93.73**	77.92	71.64	**79.49**	52.63	**90.37**	76.32	1368	73.67	127
GANet-L	**79.63**	93.67	**78.66**	71.82	78.32	53.38	89.86	**77.37**	1352	73.85	63

Table 2. Comparison with state-of-the-art methods on CULane test set. The evaluation metric is F1 score with IoU threshold=0.5. For cross scenario, only FP are shown.

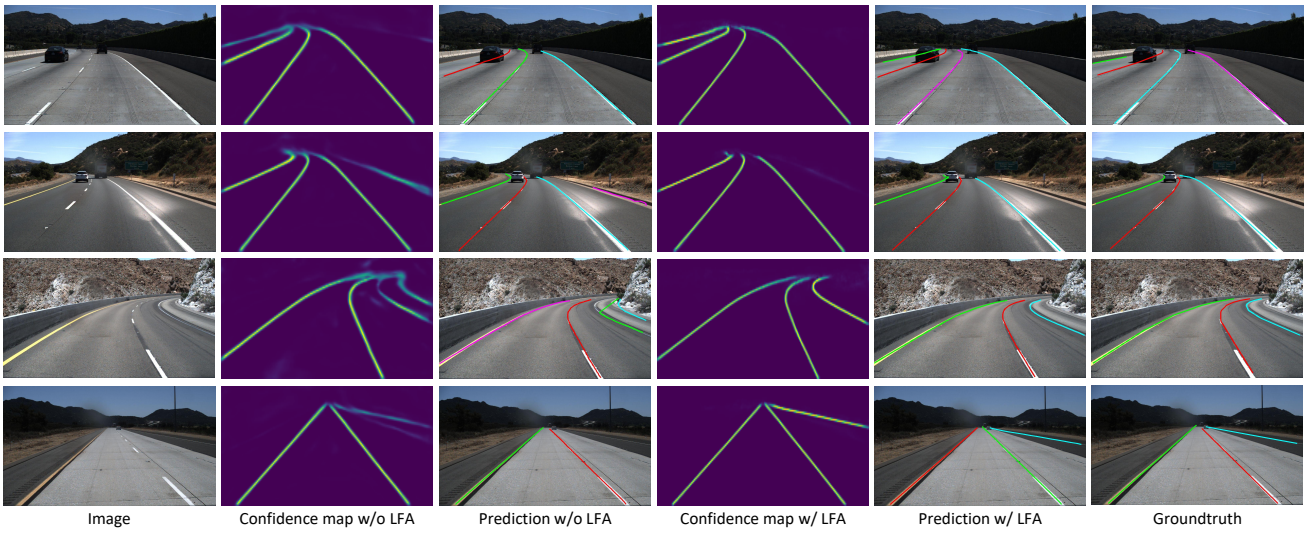

Figure 6. Visualization results of GANet w/wo LFA. The first column is the input image. The second and third columns are the predicted point confidence map and lane lines without LFA. The fourth and fifth columns are the predicted point confidence map and lane lines with LFA. The last column is the ground-truth lane lines

Baseline	LFA	AuxLoss	F1
✓			77.84
✓	✓		78.30
✓	✓	✓	78.79

Table 3. Ablation study of LFA module

LaneATT-ResNet34 which have similar speed with large margins, showing the great potential of our global association formulation. Similar to LaneATT [25], enlarging model capacity does not necessarily bring performance improvement. It is possibly because of the small amount and the single scenario of Tusimple dataset. Results have already been saturated and a larger model may cause the overfitting problem.

4.2.3 Ablation Study

To explore the properties of our proposed LFA module, we conduct an ablation study on the CULane dataset. All the

Method	F1	Acc	FP	FN	FPS
Segmentation-based					
SCNN [18]	95.97	96.53	6.17	**1.80**	7.5
EL-GAN [4]	96.26	94.90	4.12	3.36	10
ENet-SAD [7]	95.92	96.64	6.02	2.05	75
Detection-based					
FastDraw [19]	93.92	95.20	7.60	4.50	90.3
UFAST-ResNet18 [20]	87.87	95.82	19.05	3.92	312.5
UFAST-ResNet34 [20]	88.02	95.86	18.91	3.75	169.5
ERF-E2E [32]	96.25	96.02	3.21	4.28	-
LineCNN [11]	96.79	96.87	4.42	1.97	30
LaneATT-ResNet18 [25]	96.71	95.57	3.56	3.01	250
LaneATT-ResNet34 [25]	96.77	95.63	3.53	2.92	171
LaneATT-ResNet122 [25]	96.06	96.10	5.64	2.17	26
Other Methods					
PolyLaneNet [24]	90.62	93.36	9.42	9.33	115
LSTR [15]	96.86	96.18	2.91	3.38	**420**
Keypoint-based					
FOLOLane-ERF [21]	-	**96.92**	4.47	2.28	40
GANet-S	**97.71**	95.95	**1.97**	2.62	153
GANet-M	97.68	95.87	1.99	2.64	127
GANet-L	97.45	96.44	2.63	2.47	63

Table 4. Comparison with state-of-the-art methods on TuSimple test set.

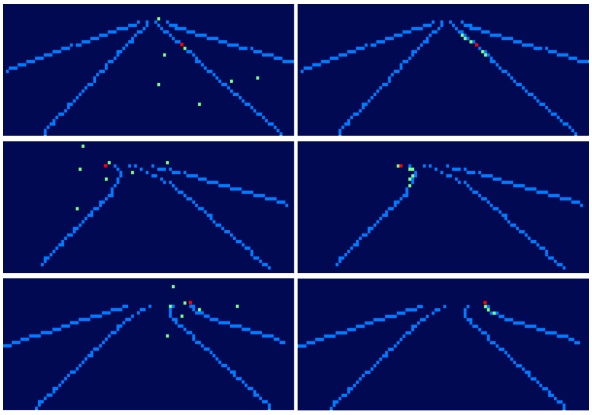

Figure 7. Visualization results of LFA w/wo auxiliary loss. The red point is the observation point. The green points are the predicted aggregation points. The light blue points are the ground-truth points on the lane line.

following experiments are based on the small version of GANet. Results are shown in Table 3. The first row shows the baseline method without our LFA module. In the second row, the LFA module is integrated into GANet without auxiliary loss. The last row shows the result of our whole GANet.

From the first two rows we can observe that LFA module without auxiliary loss is effective for lane line detection, which is due to flexible integration of context. Comparing the last two rows, we can also find that the auxiliary loss is vital to the LFA module, which can guide LFA to focus on the key information on the lane line. The visualization analysis is performed in Section 4.3.

4.3. Qualitative results

We visualize the qualitative results w/wo LFA in Figure 6. The 2-nd and 4-th columns are the visualization of confidence map without and with LFA correspondingly. As shown in the results from the first row, the LFA module makes correct prediction even with vehicle occlusion due to the fact that predicted lane points enhance each other. From the results in second and third rows, it can also be concluded that the LFA module is able to suppress background noise which may be introduced by global attention.

To intuitively investigate the properties of the LFA module, we visualize the predicted feature aggregation points in Figure 7. The first row shows a common straight lane case. With the addition of auxiliary losses, the LFA module can predict the aggregation points around the lane line. Meanwhile, the predicted aggregation points are irregular without the auxiliary loss. The last two rows show the aggregation points in the curved lane case. It is demonstrated that the LFA module is robust in its understanding of the local structures of the lane lines. This property contributes to the enhancement of lane line features and the suppression of background noise.

5. Conclusion and Discussion

In this paper, we propose a Global Association Network (GANet) to formulate the lane detection problem from a new perspective, where each keypoint is directly regressed to the starting point of the lane line instead of point-by-point extension. The association of keypoints to their belonged lane line is conducted by predicting their offsets to the corresponding starting points of lanes globally, which greatly improves the effectiveness. We further propose a Lane-aware Feature Aggregator (LFA) to adaptively capture the local correlations between adjacent keypoints to supplement local information. Experimental results show our GANet outperforms previous methods with higher speed.

Limitation. The limitation of our method is that the offsets to the starting point may become difficult to regress when the output stride is set to 1 due to the large absolute value of the offsets. In the future, we hope to address this problem by regressing the offsets with multiple levels to alleviate the regression difficulty.

6. Acknowledgments

This research is supported in part by National Natural Science Foundation of China (62022078, 62121002), National Defense Basic Scientific Research Program (JCKY2020903B002) and SenseTime Group Ltd.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 5

[2] Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Hongyang Chao, and Tao Mei. Temporal deformable convolutional encoder-decoder networks for video captioning. In *AAAI*, 2019. 3

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 3, 5

[4] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. 8

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[6] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection CNNS by self attention distillation. *ICCV*, 2019. 2

[7] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *ICCV*, 2019. 7, 8

[8] Seokwoo Jung, Sungha Choi, Mohammad Azam Khan, and Jaegul Choo. Towards lightweight lane detection by optimizing spatial embedding. *ECCVW*, 2020. 2

[9] 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

[10] 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*, 2019. 1, 3

[11] 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*, 2019. 8

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4

[14] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: A top-to-down lane detection framework based on conditional convolution. In *ICCV*, 2021. 3

[15] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *WACV*, 2021. 8

[16] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards End-to-End Lane Detection: An Instance Segmentation Approach. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018. 2

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3

[18] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*, 2018. 2, 6, 7, 8

[19] Jonah Philion. Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In *CVPR*, 2019. 7, 8

[20] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *ECCV*, 2020. 3, 7, 8

[21] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on local: Detecting lane marker from bottom up via key point. In *CVPR*, 2021. 2, 3, 6, 7, 8

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015. 1

[23] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *CVPR*, 2016. 5

[24] 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 *ICPR*, 2020. 8

[25] 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 *CVPR*, 2021. 1, 3, 7, 8

[26] TuSimple. Tusimple lane detection benchmark, 2017. https://github.com/TuSimple/tusimple-benchmark, 2017. 6

[27] 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*, 2017. 3

[28] 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 *ECCV*, 2020. 3, 7

[29] Xiangyu Xu, Muchen Li, and Wenxiu Sun. Learning deformable kernels for image and video denoising. *arXiv preprint arXiv:1904.06903*, 2019. 3

[30] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, 2019. 3

[31] Xinyi Ying, Longguang Wang, Yingqian Wang, Weidong Sheng, Wei An, and Yulan Guo. Deformable 3d convolution for video super-resolution. *IEEE Signal Processing Letters*, 2020. 3

[32] 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 Conference on Computer Vision and Pattern Recognition Workshops*, 2020. 6, 7, 8

[33] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *CVPR*, 2020. 3

[34] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 3