Supplementary Material for "Lane2Seq: Towards Unified Lane Detection via Sequence Generation"

Kunyang Zhou
Southeast University
kunyangzhou@seu.edu.cn

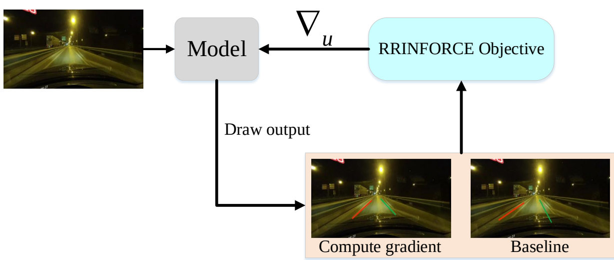

Figure 1. Illustration of REINFORCE algorithm.

1. De-quantization for different formats

Different detection formats require a specific de-quantization scheme to obtain the final predictions. A detailed description of de-quantization for each detection format is given below.

De-quantization for the *segmentation* format. we de-quantize the coordinates tokens corresponding to each polygon and then convert them into the mask. Specifically, given the x,y coordinates, their dequantization process can be expressed by $x = \frac{x}{n_{bins}} \times Wid, y = \frac{y}{n_{bins}} \times Hei$, where Wid and Hei represent the width and height of the image, respectively.

De-quantization for the *anchor* format. We directly dequantize the image coordinate tokens of the keypoints, whose dequantization process is the same as that of a polygon.

De-quantization for the *parameter* format. We dequantize the parameter tokens and vertical offset corresponding to each parameter sequence. The dequantization of parameter token can be expressed by $a_i = \frac{a_i}{n_{bins}} \times desigmoid(a_i)$, where $desigmoid$ is the inverse function of the sigmoid. The dequantization of vertical offset is $s = \frac{s}{n_{bins}} \times Hei$.

2. REINFORCE algorithm

REINFORCE algorithm [1] is a widely used method to maximize the reward function in reinforcement learning. Given an input image c, REINFORCE algorithm estimates the gradient of the reward function as below,

$$\nabla_u E_{t \sim Q}[R(t,g)] = E_{t \sim Q}[R(t,g) \nabla_u \text{log} Q(t|c,u)], \quad (1)$$

where t, and g represent the generated format-specific sequences, and ground truths, respectively. E and u denote the mathematic expectation and the model parameter. R, D, and Q stand for the reward function, data distribution of the dataset, and conditional distribution parameterized by u. In order to reduce the variance of the gradient estimate, REINFORCE algorithm usually subtracts a baseline value b from the reward function. As presented in Fig. 1, REINFORCE first draws two outputs from one training image, using one to estimate the gradient and the other to compute the baseline value. The procedure of REINFORCE algorithm is: (1) Draw two outputs from one input image. (2) Compute the reward function $R_{gradient}$ and $R_{baseline}$, whose formulation are the same as R. Final reward r is compute by $r = R_{gradient} - R_{baseline}$. (3) Estimate the gradient according to Eq. 1 and r.

3. Additional ablation studies

We conduct additional ablation experiments on the hyper-parameters. If not specified, we still carry out experiments on CULane dataset.

The size of vocabulary. We first ablate the influence of the size of vocabulary n_{bins} and the results are shown in Fig. 2. We take the segmentation format to conduct this experiment. Increasing the size of vocabulary n_{bins} can improve the model performance because the quantization error is reduced accordingly. The performance declines when n_{bins} is larger than 1000, thus we set n_{bins} to 1000.

Additional ablation Study on the Scale Factor of Different Objective Functions. We ablate the influence on the scale Factor of different objective functions for anchor and parameter format. Results are presented in Table 1 and Table 2. It can be seen that the model achieves the best performance when λ_4, λ_5, and λ_6 are 0.2, 1, and 1.5.

Ablation study on the weight of false positives of segmentation format. In Table 3, we ablate the influence of

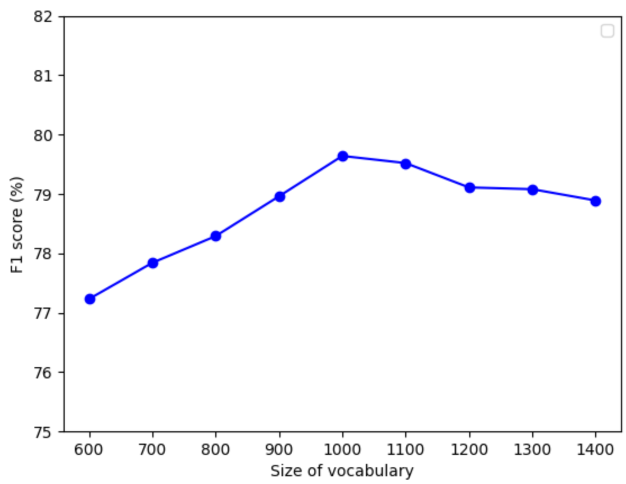

Figure 2. Influence of the size of vocabulary.

Table 1. Ablation study on different scale factors for anchor format.

λ_4	λ_5	λ_6	F1(%)↑
1	1	1	73.47
0.7	1	1	74.85
0.4	1	1	76.02
0.2	1	1	77.44
0.1	1	1	77.58
0.2	1.1	1	77.58
0.2	1.2	1	77.00
0.2	0.9	1	77.36
0.2	0.8	1	77.63
0.2	1	0.9	77.26
0.2	1	1.2	77.98
0.2	1	1.4	78.74
0.2	1	1.5	**79.27**
0.2	1	1.6	79.00

the weight of false positives of segmentation format λ_1. Performance increases when λ_1 increases, indicating introducing the penalty for false positives is beneficial. However, performance declines when λ_1 is larger than 0.3, hence we set λ_1 to 0.3.

Ablation study on the weight of false positives of anchor format. We ablate the influence of the weight of false positives of anchor format λ_2 and results are shown in Table 4. Similar to the performance trend of the segmentation format, model performance of the anchor format improves as the λ_2 increases. We set λ_2 to 0.3 according to the model performance.

Ablation study on the weight of false positives of parameter format. We further ablate the influence of the weight of false positives of parameter format λ_3 and results are shown in Table 5. It can be seen that the weight of false

Table 2. Ablation study on different scale factors for parameter format.

λ_4	λ_5	λ_6	F1(%)↑
1	1	1	74.00
0.7	1	1	74.72
0.4	1	1	76.15
0.2	1	1	77.59
0.1	1	1	77.48
0.2	1.1	1	77.48
0.2	1.2	1	77.92
0.2	0.9	1	77.85
0.2	0.8	1	77.77
0.2	1	0.9	77.53
0.2	1	1.2	78.00
0.2	1	1.4	78.15
0.2	1	1.5	**78.39**
0.2	1	1.6	77.97

Table 3. Ablation study on the weight of false positives of segmentation format.

λ_1	F1(%)↑	Precision(%)↑	Recall(%)↑
0.0	77.59	83.79	67.85
0.1	78.06	84.11	67.98
0.2	78.89	85.00	68.87
0.3	**79.64**	**85.26**	**69.00**
0.4	79.02	85.05	68.78
0.5	78.89	84.87	68.60

Table 4. Ablation study on the weight of false positives of anchor format.

λ_2	F1(%)↑	Precision(%)↑	Recall(%)↑
0.0	77.42	82.46	65.52
0.1	78.28	83.25	66.00
0.2	78.72	83.79	66.58
0.3	**79.27**	**84.72**	**67.28**
0.4	79.10	84.15	66.95
0.5	78.75	83.89	66.58

Table 5. Ablation study on the weight of false positives of parameter format.

λ_3	F1(%)↑	Precision(%)↑	Recall(%)↑
0.0	77.82	82.58	65.76
0.1	**78.39**	**83.68**	**66.94**
0.2	78.21	83.60	66.38
0.3	77.89	83.11	66.02
0.4	77.56	82.26	65.28
0.5	77.12	81.83	64.87

positives of parameter format cannot be too large. We set λ_3 to 0.1 according to the model performance.

References

[1] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 1