LATR: 3D Lane Detection from Monocular Images with Transformer Appendix

Yueru Luo[1,2] Chaoda Zheng[1,2] Xu Yan[1,2] Tang Kun[3] Chao Zheng[3]
Shuguang Cui[2,1] Zhen Li[2,1*]
[1] FNii, CUHK-Shenzhen [2] SSE, CUHK-Shenzhen [3] Tencent Map, T Lab

In this Appendix, we provide details omitted in the main paper, due to space limitations, including: a) an algorithm summarizing the dynamic plane update procedure; b) extra implementation details; c) more experimental results and visualizations.

A. Dynamic Plane Update

Here we summarize the dynamic plane update procedure (main paper Section.3.3), which is described in our main paper Section.3.3, in Algorithm. 1.

Algorithm 1 $\mathbf{PE}$ Generation and Update

Input: constructed 3D plane $\mathbf{P}$, image features $\mathbf{X}$, camera parameters T, decoder depth L.
Output:

for $l \leftarrow 0$ **to** $L-1$ **do**
 $\mathbf{P}_l^{2d} = \text{Project}\left(\mathbf{P}_l, T\right)$ $\triangleright$ <Eq.(1)>
 $\mathbf{M}_p = \text{Scatter\_Mean}(src = \mathbf{P}_l, idx = \mathbf{P}_l^{2d})$ $\triangleright$ $\mathbf{M} \in \mathbb{R}^{3\times H\times W}$, <Eq.(2)>
 $\mathbf{PE} = \text{MLP}(\mathbf{M}_p)$
 $[\Delta\theta_x, \Delta h] = \text{MLP}(\text{AvgPool}(\mathcal{G}([\mathbf{X}, \mathbf{M}_p])))$
 $\triangleright$ $\mathcal{G}$: convolution layers, <Eq.(3)>
 $\widetilde{\mathbf{P}}_{l+1}^{\mathsf{T}} = D \cdot \widetilde{\mathbf{P}}_l^{\mathsf{T}}$ $\triangleright$ $\mathbf{D}$ is from Eq.(4)
 $\triangleright$ update plane $\mathbf{P}$, <Eq.(5)>
end for

B. Implementation Details

Set Prediction. We cast the 3D lane detection (Sec.3.5) and the auxiliary instance segmentation (Sec.3.2) as a set prediction problem. To enable consistent label assignment for both tasks (Sec.3.5), we adopt a dice-based matching score for label assignment, introduced in [3]. The ground truths for each input are padded with $\varnothing$ up to N, where N indicates the prediction lane count. For i-th prediction and j-th ground truth, their matching score is defined as:

$$\mathcal{S}_{i,j} = p_{i,c_j}^{1-\alpha} \cdot dice(\hat{m}_i, m_j)^{\alpha} \tag{1}$$

where c_j denotes the category label for j-th ground-truth lane. p_{i,c_j} denotes the probability for the category c_j of i-th prediction. $\hat{m}_i$ and m_j represent the masks of i-th prediction and j-th ground-truth lane respectively. $dice$ is the dice coefficients [6] of segmentation masks. The weight of α balances the importance of classification and segmentation tasks and is set to 0.8, as in [3]. Then the optimal one-to-one matching is acquired using the Hungarian algorithm [7].

Auxiliary Loss. Our auxiliary segmentation loss comprises three components: objectness, segmentation, and classification, organized as:

$$\begin{aligned} \mathcal{L}_{aux} &= \lambda_o \cdot \mathcal{L}_{obj} + \mathcal{L}_{mask} + \lambda_c \cdot \mathcal{L}_{cls}, \\ \mathcal{L}_{mask} &= \lambda_{dice} \cdot \mathcal{L}_{dice} + \lambda_{bce} \cdot \mathcal{L}_{bce} \end{aligned} \tag{2}$$

where $\mathcal{L}_{obj}$ is a BCELoss for objectness, $\mathcal{L}_{cls}$ is focal loss [5] for lane classification that uses the same γ and α as the 3D lane classification (Sec.3.5), and $\mathcal{L}_{mask}$ is composed of the dice loss [6] and pixel-wise BCELoss for the foreground and background balance purpose. In our experiments, we adopt the following setup: $\lambda_o = 1.0$, $\lambda_c = 2.0$, $\lambda_{dice} = 2.0$ and $\lambda_{bce} = 5.0$.

C. More Experiment Results

We further conduct more ablation studies of the architecture on OpenLane-300 *val* set, aligning with the main paper ablation setting.

C.1. More Ablations

Number of Decoder Layers. Our LATR uses the Transformer decoder for 3D lane detection. By varying the number of decoder layers, we found that increasing the number of layers from 2 to 8 improved the F1 score from 68.7 to 70.9 (Tab. 1). We use a six-layer decoder as the default, following [1, 4]. However, we observed that using a two-layer decoder achieves comparable performance to the default six-layer version, leading us to adopt a two-layer decoder as our lite version LATR-Lite for higher efficiency. Notably, our LATR-Lite significantly improves efficiency and effectiveness compared to the previous SoTA, as shown

*Corresponding author

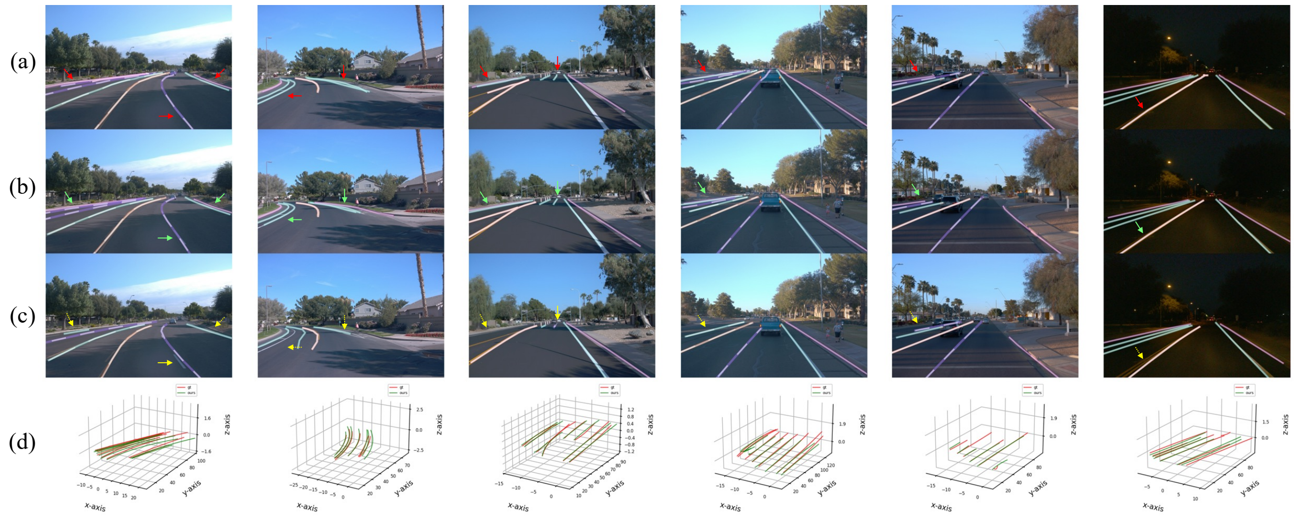

Figure 1: **More visualizations on OpenLane *val* set.** Rows (a), (b), (c) illustrate the ground truth lanes, predictions generated by LATR and Persformer [2], respectively. The lanes are projected onto 2D images and different colors in the images represent specific categories in OpenLane. Missed lanes are indicated by arrows with dashed lines, while existing lanes are indicated by arrows with solid lines. In row (d), we compare the ground truths (red) and the predictions of our LATR (green) in 3D space. Best viewed in color (zoom in for details.)

in Tab. 2. While employing nearly half the parameters, LATR-Lite not only achieves a twofold increase in FPS but also exhibits a notable enhancement in F1 score, with an impressive improvement of +8.5.

# Layers	F1 / *C.Acc.*	X error (m) *near* | *far*	Z error (m) *near* | *far*
2	68.7 / 90.9	0.260 | 0.324	0.097 | 0.130
4	69.9 / 92.3	0.257 | 0.325	0.097 | 0.133
6	70.4 / 92.9	0.241 | 0.321	0.097 | 0.132
8	70.9 / 93.0	0.257 | 0.329	0.098 | 0.134

Table 1: **Ablation study on number of decoder layers.**

Input Sizes. Image resolution is a key factor that influences performance. To study the impact of different input shapes, we compared four resolutions, as detailed in Tab 3. Notably, the F1 score demonstrates improvement with increasing input size. This observation is consistent with our intuitive expectations, as larger images containing finer details can enhance the accuracy of lane location detection.

C.2. Model Complexity.

To comprehensively evaluate the performance of our proposed LATR, we compared its model parameters and FPS with those of the previous state-of-the-art model [2], as shown in Tab 2. Experimental results reveal that our model, LATR, achieves a superior F1 score of 61.9 and a frame rate of 11.34 FPS, outperforming Persformer, which exhibits a lower F1 score of 53.0 and operates at 6.92 FPS. Notably, our LATR-Lite significantly improves efficiency and effectiveness compared to Persformer, despite using almost half the number of parameters. Specifically, we achieve a more than twofold increase in FPS, while obtaining a remarkable +8.5 improvement in F1 score. The F1 results are compared on OpenLane *val* set.

Model	Backbone	# Params	FPS	GPU Cost	F1
360×480					
Persformer	Efficient-B7	54.94 M	11.48	2.16GB	50.5
Persformer	Res50	62.54 M	9.96	2.24GB	52.6
LATR	Res50	44.35 M	12.63	2.28GB	58.6
720×960					
Persformer	Res50	63.19 M	6.92	3.00GB	53.0
LATR-Lite	Res50	38.78 M	17.75	2.26GB	61.5
LATR	Res50	44.35 M	11.34	2.55GB	61.9

Table 2: **Model complexity.** All models are tested on single A100 GPU and AMD EPYC 7351@2.60GHz CPUs. The reported F1 scores are based on OpenLane *val* set, aligning with the main results in the main paper.

Input Size	F1 / *C.Acc.*	X error (m) *near* | *far*	Z error (m) *near* | *far*
360×480	63.8 / 90.4	0.310 | 0.384	0.113 | 0.161
480×640	67.4 / 92.1	0.262 | 0.346	0.100 | 0.140
720×960	70.4 / 92.9	0.241 | 0.321	0.097 | 0.132
960×1280	71.2 / 92.9	0.253 | 0.315	0.096 | 0.129

Table 3: **Ablation study on input size.**

C.3. Visualizations

We present additional qualitative analysis in Fig. 1, highlighting differences using arrows of different colors. This analysis demonstrates that LATR can produce more accurate and robust 3D lane results.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 1

[2] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[3] Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, and Wenyu Liu. Sparse instance activation for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4433–4442, 2022. 1

[4] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022. 1

[5] 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. 1

[6] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 1

[7] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016. 1