SparseLaneSTP: Leveraging Spatio-Temporal Priors with Sparse Transformers for 3D Lane Detection

Supplementary Material

1. More details on the 3D lane representation

In this section, we provide additional information about CR splines and the curve computation.

1.1. CR splines

We provide details on our selected curve representation, Catmull-Rom (CR) splines [1]. CR splines are a class of splines comprised of piece-wise defined smooth third-order polynomial splines. A special property that we exploit for our transformer-tailored representation is that the curve inherently passes through its control points. In Fig. 1a we illustrate the basis functions for an example CR spline using 6 control points. Each value of a CR spline function is affected by the 4 nearest control points. Thus, a local segment between two consecutive control points p_j and p_{j+1} of a 1D spline $f(\tilde{s})$ with curve argument $\tilde{s} \in [0,\, 1]$ is defined as

$$f(\tilde{s}) = \begin{bmatrix}\tilde{s}^3 & \tilde{s}^2 & \tilde{s} & 1\end{bmatrix} \cdot \tilde{\mathbf{M}}_{\text{CR}} \cdot \begin{bmatrix} p_{j-1} \\ p_j \\ p_{j+1} \\ p_{j+2}\end{bmatrix} \tag{1}$$

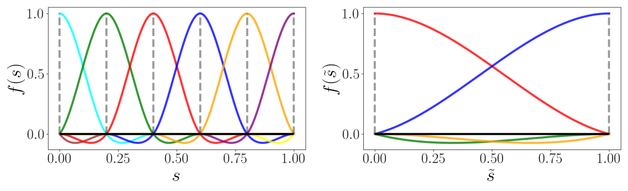
and illustrated in Fig. 1b. $\tilde{\mathbf{M}}_{\text{CR}} \in \mathbb{R}^{4\times4}$ is the coefficient matrix for this segment defined as

$$\tilde{\mathbf{M}}_{\text{CR}} = \frac{1}{2} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{2}$$

and $\tilde{\mathbf{s}} = \begin{bmatrix}\tilde{s}^3 & \tilde{s}^2 & \tilde{s} & 1\end{bmatrix}$ the spline argument vector. The product of $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{M}}_{\text{CR}}$ represents the four involved CR basis polynomials. The multiplication with the control point vector yields a weighted sum, which defines the shape of the curve segment.

To extend this formulation from a local segment to a global curve with an arbitrary number of control points M, we have to apply two changes. First, the spline argument vector needs to be normalized based on the knot positions s_k, which are visualized in Fig. 1a by the dashed vertical lines, such that we obtain

$$\mathbf{s} = \begin{bmatrix} \big((s-s_k)/(s_{k+1}-s_k)\big)^3 \\ \big((s-s_k)/(s_{k+1}-s_k)\big)^2 \\ (s-s_k)/(s_{k+1}-s_k) \\ 1 \end{bmatrix}^T \tag{3}$$



(a) CR basis polynomials (b) Local segment

Figure 1. Global and local CR spline basis functions.

for $s \in [s_k,\, s_{k+1}]$. Second, the coefficient matrix needs to be augmented to $\mathbf{M}_{\text{CR}} \in \mathbb{R}^{4\times M}$, where most values remain zero and only the entries affecting the four involved control points equal $\tilde{\mathbf{M}}_{\text{CR}}$. Thus, $\mathbf{M}_{\text{CR}}(s)$ now varies with the curve parameter s and is determined by the interval between the corresponding knots s_k and s_{k+1}. Consequently, we obtain

$$f(s) = \mathbf{s} \cdot \mathbf{M}_{\text{CR}}(s) \cdot \mathbf{p} \quad \text{with} \tag{4}$$

$$\mathbf{M}_{\text{CR}}(s) = \begin{cases} \tilde{\mathbf{M}}_{\text{CR}} & \text{if } s \in [s_k, s_{k+1}] \\ 0 & \text{else}\,, \end{cases} \tag{5}$$

with control point vector $\mathbf{p} \in \mathbb{R}^M$. Note, that although $\mathbf{M}_{\text{CR}}(s)$ depends on s, which we omitted in the main paper for simplicity, the matrix $\mathbf{s} \cdot \mathbf{M}_{\text{CR}}(s)$ can be pre-computed before training or inference. Thus, in each training iteration or inference step, the post-processing required to generate full splines reduces to a single matrix multiplication.

Finally, extending the 1D function to a full curve model $\mathbf{f}(s) \in \mathbb{R}^4$ with 3 spatial $(x_j,\, y_j,\, z_i)$ and 1 visibility (v_j) results in

$$\mathbf{f}(s) = \begin{bmatrix}s^3 & s^2 & s & 1\end{bmatrix} \cdot \mathbf{M}_{\text{CR}}(s) \cdot \mathbf{P} \quad \text{with} \tag{6}$$

$$\mathbf{P} = \begin{bmatrix} x_1 & y_1 & z_1 & v_1 \\ & \cdots & & \\ x_j & y_j & z_j & v_j \\ & \cdots & & \\ x_M & y_M & z_M & v_M \end{bmatrix}, \tag{7}$$

with control point matrix $\mathbf{P} \in \mathbb{R}^{M\times4}$ and $\mathbf{M}_{\text{CR}}(s)$ the full coefficient matrix.

1.2. Curve computation and parameterization

In practice, the curve is computed by sampling arguments s in the range $[0,\, 1]$, pre-computing the matrix $\mathbf{s} \cdot \mathbf{M}_{\text{CR}}(s)$ and generating the curve via multiplication with the control point matrix (as described in Section 1.1).

During training, we need to determine the curve argument $\hat{s}$ for a given ground truth sample $(\hat{x}, \hat{y}, \hat{z}, \hat{v})$ in order to compute the loss at the corresponding position in the predicted curve. Similar to [7], we keep the longitudinal y-component of the spline control points fixed and distribute them uniformly along the range $[y_s, y_e]$. This simplifies the determination of curve arguments $\hat{s}$ to

$$\hat{s} = (\hat{y} - y_s)/(y_e - y_s)\,. \tag{8}$$

2. Detection losses

In this section, more details about our utilized losses are provided.

2.1. Classification loss

We employ focal loss [5] for category classification, formulated as follows

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K+1}\left(\hat{\mathbf{C}}_{ik}\cdot\left(1-\mathbf{C}_{ik}\right)^{\gamma}\cdot\log\left(\mathbf{C}_{ik}\right)\right), \tag{9}$$

with predicted category probability distribution of the i^{th} line proposal $\mathbf{C}_i$, ground truth one-hot vector $\hat{\mathbf{C}}_i$ and focusing parameter $\gamma \geq 0$. Note, that the probabilities for predicted categories and the background class sum up to $\sum_{k=1}^{K+1}\mathbf{C}_{ik} = 1$.

2.2. Regression loss

Inspired by [8], we apply L1 loss along the visible points

$$\mathcal{L}_{reg} = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{M_{\text{GT}}}\hat{v}_{ij}\cdot\left|\left|\begin{pmatrix}\mathbf{f}_{x,i}(\hat{s}_j)\\ \mathbf{f}_{z,i}(\hat{s}_j)\end{pmatrix}-\begin{pmatrix}\hat{x}_{ij}\\ \hat{z}_{ij}\end{pmatrix}\right|\right|_1, \tag{10}$$

with ground truth points $(\hat{x}_{ij}, \hat{y}_{ij}, \hat{z}_{ij}, \hat{v}_{ij})$ of the i^{th} line proposal and M_{GT} the number of samples in the ground truth. The corresponding curve argument $\hat{s}_j$ is obtained as described in Section 1.2.

2.3. Visibility loss

For the visibility, we use binary cross-entropy

$$\mathcal{L}_{vis} = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{M_{\text{GT}}}\hat{v}_{ij}\cdot\log\left(\mathbf{f}_{v,ij}(s_j)\right) + \tag{11}$$

$$(1-\hat{v}_{ij})\cdot\log\left(1-\mathbf{f}_{v,ij}(s_j)\right). \tag{12}$$

3. Spatio-temporal attention

As discussed in the main paper, the three components of the spatio-temporal attention mechanism interact with distinct subsets of queries, which serve as keys and values. This is implemented through masked attention, where a dedicated mask is computed for each component – SLA, PNA and TCA – based on either the structure of the query matrix or the positions of the associated control points.

3.1. Masking

In SLA, for a given input query $\tilde{\mathbf{Q}}_{ij}$, the corresponding set of keys and values, $\tilde{\mathbf{Q}}_{\text{SLA}}$, consists of queries from the same line proposal i. Thus, each query in SLA interacts with the M queries on the same line.

In PNA, points on neighboring lines in the orthogonal direction to the curve are selected. To achieve this, we employ the method described in [8] to identify the two nearest neighbors on each adjacent line in the orthogonal direction. Thus, each query in PNA interacts with the $2 \cdot (N-1)$ parallel neighbors.

In TCA, queries in the memory associated with the M_{TCA} nearest temporally propagated control points are selected. For the distance metric we use euclidean distance. Thus, each query in TCA interacts with the M_{TCA} nearest queries in the memory queue. Note, that we used $M_{\text{TCA}} = 10$ in our experiments.

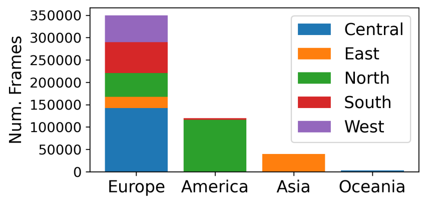
3.2. Potential for efficiency benefits

We also note that our approach has potential for further efficiency improvements, which could be explored as future work: In its core, the proposed spatio-temporal attention module masks out redundant relations of tokens. The resulting attention matrix has only $\sim 10\,\%$ of active token relations and thus $\sim 90\,\%$ sparsity. While our implementation is currently based on a simplistic standard attention-layer that uses masking without leveraging sparsity, optimized approaches like FlashAttention [3, 4] could be used in future in combination with deployment on more modern GPUs. Such an implementation could exploit the large extent of sparsity in the attention-layers and potentially lead to severe savings in runtime and memory.

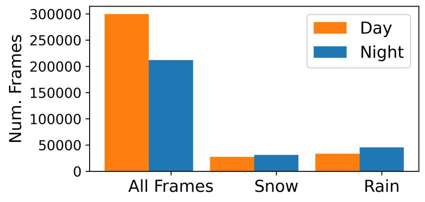
4. Dataset statistics

Fig. 2 shows several statistics of our dataset with respect to the driving scenario and environment. The sequences were recorded in various regions and countries across the globe with the distribution given in Fig. 2a. We made sure to collect the data under various weather conditions and daytimes (Fig. 2b). Moreover, the sequences were chosen such that the proportions of highway, urban and rural situations are approximately equal and made sure that even corner case situations like construction zones are included (Fig. 2c). Finally, Fig. 2d demonstrates that the dataset also covers lanes from high curvature roads to enable the trained model to learn from challenging driving scenarios.

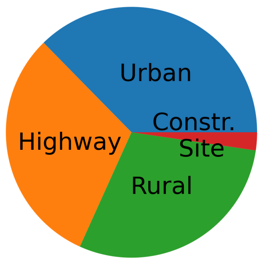
In Fig. 3 lane marking specific statistics are illustrated. From Fig. 3a is is obvious that the number of occurring lane markings per frame varies across our dataset. The highest count of frames is shown for a number of 5 markings per frame and the maximum number of occurring lines is 11. In Fig. 3b we show the distribution of line categories. Be-



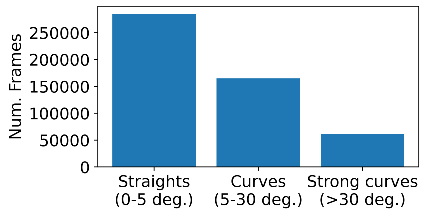
(a) Regions and countries



(b) Weather and daytime

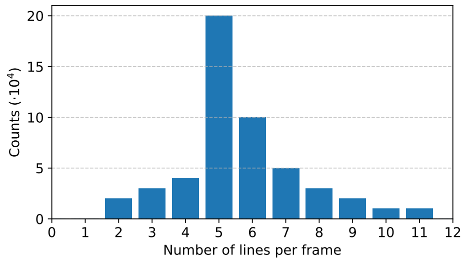


(c) Environments

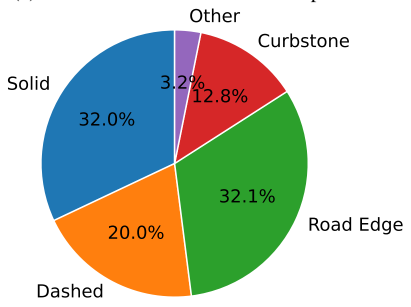


(d) Lane curvature (according to accumulated degree)

Figure 2. Scenario-based dataset statistics.



(a) Distribution of number of lines per frame



(b) Line category distribution

Figure 3. Lane marking specific dataset statistics.

sides the frequently occurring dashed and solid markings, the labels also contain a large number of road edges and even curbstones. “Other” contains special kinds of line-like objects like botts’ dots and road furniture.

5. Auto-labeling details

In the main paper we already gave an overview of the components and functionality of our auto-labeling pipeline. More details regarding the most important components are provided in the following.

2D lane detector. A profound 2D lane detector based on LaneATT [9] is utilized to generate 2D pseudo labels, which are elevated to 3D space in a later stage. Certain modifications were applied to the model, which lead to slight improvements and more robust detection behavior particularly in the near-range detection. The network was further trained using an additional set of 2D labeled images obtained from recordings captured with the same vehicles and cameras. Subsequently, the model is deployed to perform the 2D pseudo-labeling task on the unlabeled set of sequences in our dataset. For a more detailed assessment of the performance of the 2D lane detector, we refer to the qualitative and quantitative results of the experiment section from [9].

3D surface model. We use a simple yet sufficient road surface model based on the precise ego-trajectory of the vehicle. The assumption of the road model is that the orientation of the vehicle, which is given in the ego-trajectory, spans a local plane-based surface segment around each of its locations. Each plane segment is defined between the current and next position in the ego-trajectory, resulting in a first degree spline surface in 3D space. Since the ego-trajectory, which defines this surface model, was precisely optimized using visual odometry approaches, it provides an accurate estimation of the real road surface in the vicinity of the vehicle. Since we propose to project only the first segments of 2D pseudo-labels to this surface, the resulting points in 3D space will usually not have a large distance to the vehicle. The 3D position of the projected lane points can therefore be considered sufficiently reliable.

Projection to 3D. Since the surface model consists merely of stacked planes, the intersection of a visual ray (originating in the camera center traveling through a 2D lane point) and the road surface can be analytically computed. Thus, the projection of 2D lane points onto the road surface is a simple and low-cost operation.

Accumulation and Tracking. Connecting line points in

3D to obtain global line instances requires tracking. For the tracker a solution based on the prominent Kalman Filter is utilized with certain modifications to get a tailored solution for the line tracking task (instead of standard object tracking). Applying the line tracker to the 3D line points, we receive full line instances with consistent track IDs.

3D ground truth per frame. After accumulation and tracking, the resulting global line instances, which are defined for an entire sequence, are simply transformed to the local vehicle coordinate systems for each frame in the sequence utilizing the ego-pose information for the respective frame. As a result we obtain 3D lines for each frame in its local 3D coordinate system.

6. Additional results

We provide additional qualitative comparisons of our model SparseLaneSTP and LATR [6].

6.1. Qualitative results on OpenLane

Fig. 4 shows additional qualitative results from our model compared to LATR. These examples emphasize the advantages of our proposed contributions discussed in the main paper while also exploring new scenarios in greater detail.

Fig. 4a demonstrates the advantage of our continuous representation in precise start-point estimation. Additionally, Fig. 4b highlights another key benefit. The network inherently produces smooth curves, whereas the discrete lane model struggles to accurately represent the sharp continuous curve on the right.

Fig. 4c illustrates a case where our model accurately detects the two splitting lines, whereas LATR fails. This suggests that our novel STA component effectively focuses on neighboring points of both lines, whereas LATR relies on global attention, incorporating a majority of redundant relations. Additionally, the applied spatial regularization enforces smoothness, penalizing predictions similar to those produced by LATR.

Fig. 4d and Fig. 4e illustrate the advantages of temporal modeling over the non-temporal LATR approach. In Fig. 4d, the crossing vehicle further obstructs the already faint lane markings. Nonetheless, our model maintains consistent detections. A similar behavior is observed in Fig. 4e, where partial occlusion does not impact detection performance.

An interesting case is shown in Fig. 4f, depicting a scenario with severely limited visibility due to extreme weather conditions. With most of the image blurred, LATR fails and produces random predictions. In contrast, our model leverages queries stored in memory, enabling stable and sufficient detection.

6.2. Qualitative results on our 3D lane dataset

We also compare our method to the baseline on our 3D lane dataset. Note that the top-view and height plots in Fig. 5 depict twice the range of those in Fig. 4 while being scaled to the same width for consistency.

Fig. 5a demonstrates that our model accurately detects occlusions and even captures small visible segments, whereas the baseline using the discrete representation fails to represent such fine details.

As shown in Fig. 5c, our method consistently captures the long-range curve within the visible area, whereas the baseline exhibits inaccurate regression and visibility estimation due to simple extrapolation. Similarly, in Fig. 5d and Fig. 5e, our model maintains consistent detection throughout the sequence - despite reflections caused by poor lighting conditions. Notably, Fig. 5e presents a highly challenging scenario, where our model still achieves reasonably accurate results.

An interesting case is presented in Fig. 5d. At first glance, the baseline appears to yield better results, however, the top-view and height profile reveal inaccuracies in the 3D geometry due to the absence of priors. In contrast, our method, which incorporates spatial priors, demonstrates a more accurate understanding of lane structure, likely attributed to the STA mechanism.

6.3. Scenario-based quantitative comparison on our 3D lane dataset

Given the variety of our dataset with respect to driving environment, daytime, weather and curvature as illustrated in Fig. 2, we split the test set into eleven different scenario subsets and include a quantitative comparison of our model to the other methods for each subset in Table 1.

From the comparison it is clear that our model outperforms the other methods for each scenario. Notably, the margin is even larger for curves and strong curves than for straights, highlighting the capability of SparseLaneSTP to accurately detect lanes in scenarios that show challenging road geometries.

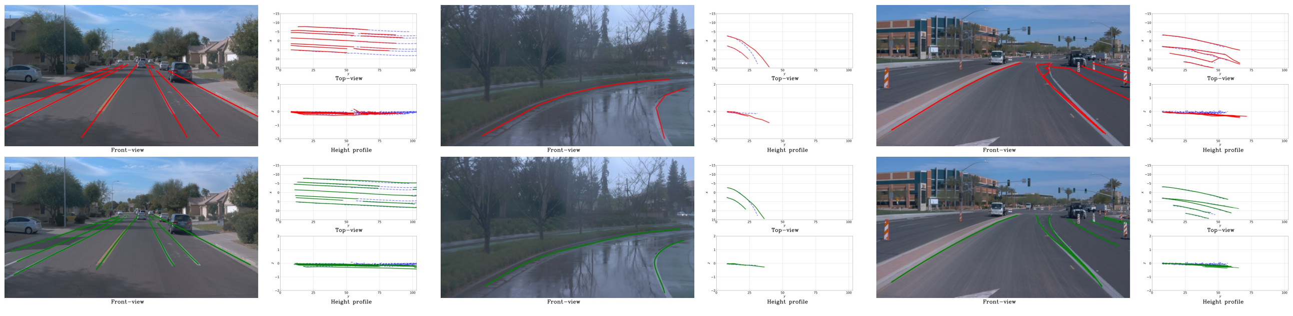
Besides, the benefit of spatio-temporal priors becomes evident in scenarios of poor visibility (rain, fog). Here the model apparently leverages prior knowledge about road structure and / or previous predictions and queries instead of suffering under poor signal due to the bad view of single frames, leading to a more robust detection behavior. Note that situations in the snow test subset, where the gap of F1-Score is less significant, do not necessarily imply poor visibility since "Snow" does not correspond to precipitation but only to snowy environments.

References

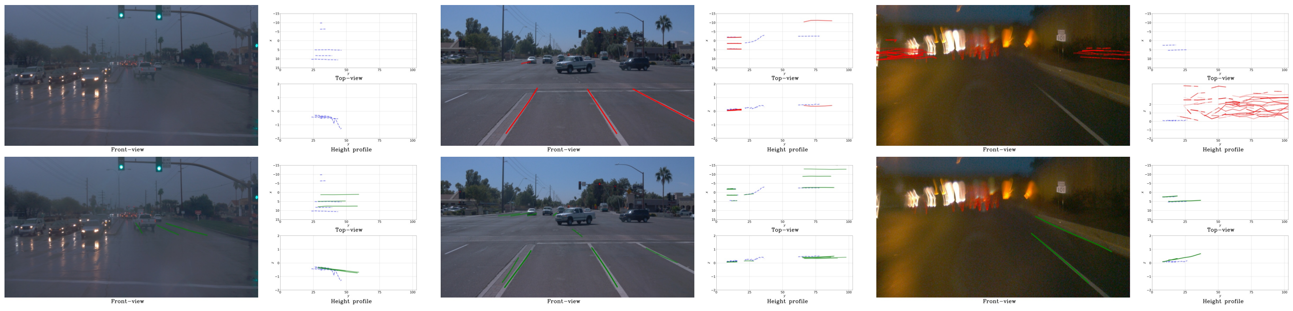
[1] Edwin Catmull and Raphael Rom. A class of local interpolating splines. *Computer aided geometric design*, 1972. 1

Method	Environment			Daytime / Weather					Curvature		
	Highway	Urban	Rural	Sun	Night	Rain	Snow	Fog	Straights	Curves	Strong Curves
PersFormer [2]	64.2 %	48.5 %	60.7 %	61.2 %	55.1 %	52.6 %	52.9 %	54.3 %	64.4 %	46.8 %	22.1 %
LaneCPP [8]	67.9 %	51.4 %	65.1 %	64.0 %	58.9 %	55.4 %	56.3 %	58.1 %	70.2 %	50.8 %	26.3 %
Baseline (LATR) [6]	70.4 %	53.8 %	66.3 %	66.5 %	60.2 %	57.7 %	58.7 %	59.3 %	73.0 %	51.6 %	29.8 %
SparseLaneSTP (ours)	**74.6 %**	**58.0 %**	**72.2 %**	**70.6 %**	**64.7 %**	**64.9 %**	**61.5 %**	**66.6 %**	**75.8 %**	**59.5 %**	**35.3 %**

Table 1. F1-Score comparison for different scenarios on our dataset. **Best performance** and second best are highlighted.



(a) (b) (c)



(d) (e) (f)

Figure 4. Qualitative comparison of LATR and SparseLaneSTP on OpenLane with ground truth for reference.

[2] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 5

[3] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv:2307.08691*, 2023.

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems (NeuIPS)*, 2022.

[5] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[6] Yueru Luo, Chaoda Zheng, Xu Yan, Tang Kun, Chao Zheng, Shuguang Cui, and Zhen Li. Latr: 3d lane detection from monocular images with transformer. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 4, 5

[7] Maximilian Pittner, Alexandru Condurache, and Joel Janai. 3d-splinenet: 3d traffic line detection using parametric spline representations. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2

[8] Maximilian Pittner, Joel Janai, and Alexandru P Condurache. Lanecpp: Continuous 3d lane detection using physical priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5

[9] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

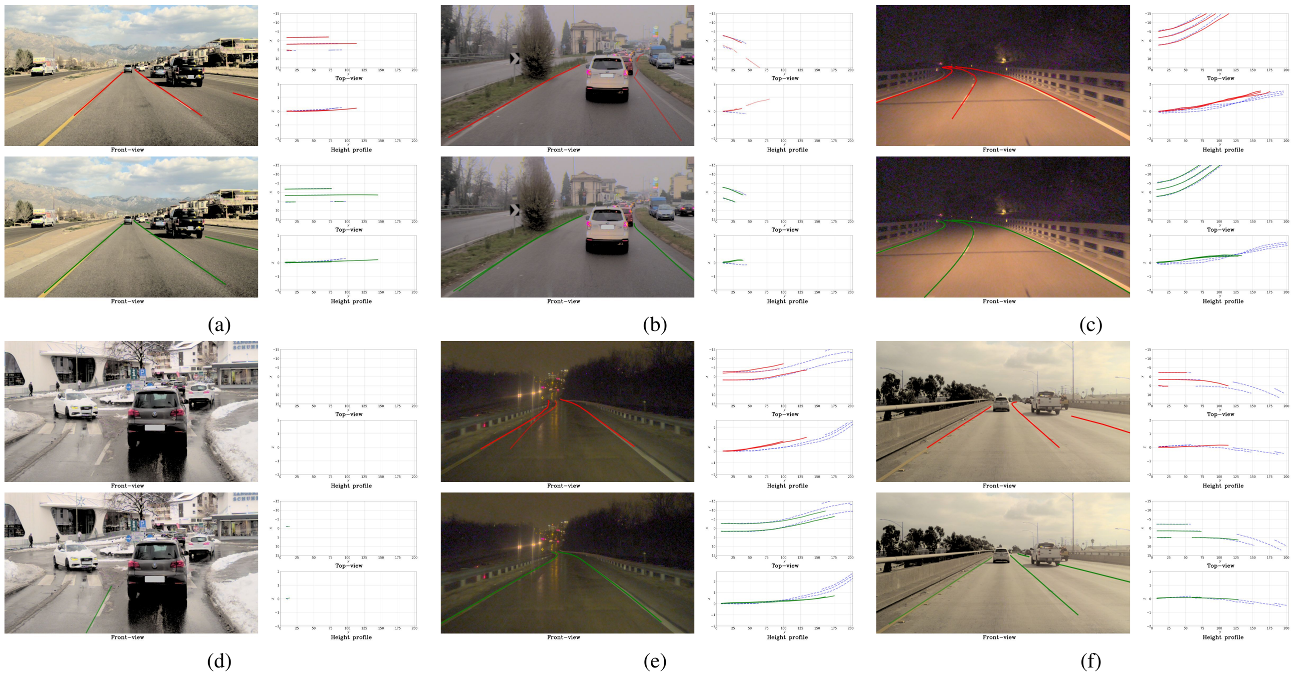


Figure 5. Qualitative comparison of LATR and SparseLaneSTP on our dataset with ground truth for reference.