Supplementary Material for MamV2XCalib: V2X-based Target-less Infrastructure Camera Calibration with State Space Model

Yaoye Zhu Zhe Wang Yan Wang*
Institute for AI Industry Research (AIR), Tsinghua University
zhuyaoye22@gmail.com wangzhe@air.tsinghua.edu.cn wangyan@air.tsinghua.edu.cn

1. Datasets

We conducted experiments on the V2X-Seq [11] and TUMTraf-V2X [15] datasets. On V2X-Seq [11] dataset, the training set consists of approximately 7500 pairs of time-aligned point clouds and roadside camera images. The validation set contains roughly 3300 pairs of data. The test set includes around 4000 pairs. There is no overlap among the three subsets. On TUMTraf-V2X [15] dataset, we have more than 2000 pairs to train our model and 300 pairs to test the effect.

2. Calibration Flow

Flow fields are often used to describe the motion of points in space. For example, optical flow represents the movement of image pixels over a time series, describing the motion vector field between two consecutive frames. Similarly, scene flow characterizes object motion in 3D space, estimating the motion vector field for each 3D point [12]. Inspired by these concepts, we introduce the notion of calibration flow, which describes the vector field between two pixels corresponding to the same 3D point after projection onto a plane through different coordinate transformations T_1, T_2. From its definition, the calibration flow has a direct relationship with the coordinate transformations. The mathematical expression is as follows:

$$d_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{KT} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{1}$$

where the meanings of the letters are consistent with the main text of the paper. We obtain the projected points through T_1, T_2 and Eq. 1. In the image coordinate system, the coordinates are represented as $[u, v]^T$ and $[\tilde{u}, \tilde{v}]^T$. The calibration flow (p, q) for pixel $[u, v]^T$ is defined as:

$$p = \tilde{u} - u, \quad q = \tilde{v} - v \tag{2}$$

*Corresponding author: wangyan@air.tsinghua.edu.cn

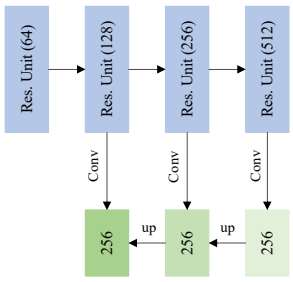


Figure 1. Multi-scale feature extraction block.

The projected depth map D_1 of the vehicle point cloud is obtained through the real coordinate transformation T_{LC}. Theoretically, for the same spatial point in the real world, the pixels of depth map D_1 should overlap with the pixels of the image I. Therefore, we treat the known image as a substitute for the unknown depth map D_1. Additionally, the point cloud can produce depth map D_2 through the known initial coordinate transformation T_{init}. We estimate the calibration flow based on the image I and D_2 to regress the extrinsic parameters.

3. Implementation Details

Our network was implemented using PyTorch 2.1.1, with CUDA 11.8 serving as the underlying acceleration framework. All training and evaluation processes were carried out on a single NVIDIA A100 GPU equipped with 40GB of memory.

Multi-scale feature extraction: We use a multi-scale ResNet-18 as the feature extractor. The structure is shown in Fig. 1, and the final output feature map has dimensions of $256 \times 32 \times 64$.

Feature query and Iterative update: We utilize a Lookup operation [9] to extract features useful for extrinsic parameter estimation from a 4D correlation volume that contains comprehensive pixel correspondence information. As shown in Fig. 2, for each pixel a, its corresponding pixel b can be determined based on the calibration flow. We ag-

Method	Range	X(cm)	Y(cm)	Z (cm)	roll($^\circ$)	pitch ($^\circ$)	yaw ($^\circ$)
CalibRCNN [8]	$[-0.25m, 0.25m]$ / $[-10^\circ, 10^\circ]$	6.20	4.30	5.40	0.20	0.64	0.45
Calibformer [10]	$[-0.25m, 0.25m]$ / $[-10^\circ, 10^\circ]$	1.10	0.90	1.56	0.08	0.26	0.09
CalibDepth [13]	$[-1.5m, 1.5m]$ / $[-20^\circ, 20^\circ]$	1.31	1.02	1.17	0.06	0.23	0.08
LCCNet [6]	$[-1.5m, 1.5m]$ / $[-20^\circ, 20^\circ]$	0.26	0.36	0.35	0.02	0.11	0.03
CMRNext [1]	$[-1.5m, 1.5m]$ / $[-20^\circ, 20^\circ]$	1.12	0.83	0.79	0.04	0.04	0.04
Ours	$[-1.5m, 1.5m]$ / $[-20^\circ, 20^\circ]$	0.25	0.41	0.87	0.02	0.13	0.04

Table 1. Comparison of calibration results applied to the single-Vehicle dataset KITTI-odometry [3].

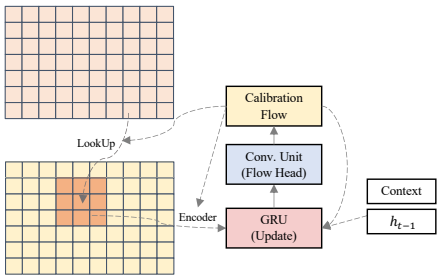


Figure 2. Query features from the volume and iteratively update.

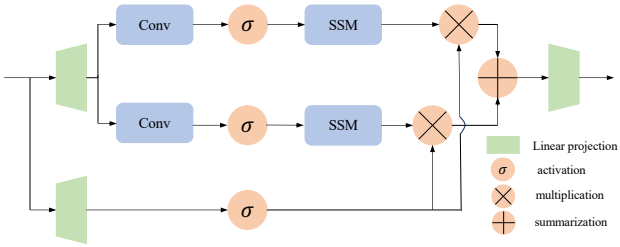


Figure 3. Bidirectional mamba architecture.

gregate similarity values within a radius r around the pixel b. r is set to 4 in our network.

Bidirectional Mamba Block: The bidirectional Mamba architecture consists of forward and backward modules [2, 14]. The mathematical definition of a single Mamba module is as follows [4]:

$$\begin{aligned} h'(t) &= Ah(t-1) + Bx(t) \\ y(t) &= Ch(t) \end{aligned} \tag{3}$$

The continuous system is discretized here using zero-order hold (ZOH).

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \end{aligned} \tag{4}$$

$$\begin{aligned} h_t &= \bar{A}\bar{h}_{t-1} + \bar{B}\bar{x}_t \\ y_t &= C\bar{h}_t \end{aligned} \tag{5}$$

In mamba, a more efficient convolution mode is introduced, which bypasses the state computation and implements a convolution kernel.

$$\begin{aligned} \bar{K} &= (C\bar{B}, C\bar{A}\bar{B}, ..., C\bar{A}^k\bar{B}, ...) \\ \bar{y} &= \bar{x} * \bar{K} \end{aligned} \tag{6}$$

where $\bar{K}$ is a structured convolutional kernel.

Further, we adopt a bidirectional Mamba design in our network, as shown in Fig. 3. As described in our paper, after obtaining a series of calibration flows, we divide the calibration flow into patches for input. Specifically, we use a 5-frame image sequence and iterate 10 times, resulting in 50 calibration flow maps. The patch size is set to 16×16.

4. More Experimental Results

In this section, we present additional experimental results.

4.1. Directly transfer to the single-vehicle dataset

Our method can naturally transfer to the single-vehicle LiDAR-camera calibration problem. As mentioned earlier, in single-vehicle scenarios, translation deviation is non-negligible relative to the LiDAR-camera distance. Therefore, we incorporate a translation error estimation at the output of the original network, and the results are shown in Tab. 1. The results show that although our method was originally designed to address roadside camera calibration in V2X scenarios, it remains competitive in single-vehicle scenarios.

4.2. Robustness Under Slight Translation Noise

We further demonstrate through experiments that our method can accurately calibrate rotational deviations in V2X scenarios even in the presence of slight translation, as shown in Tab. 2.

Initial Translation Error (cm)	Mean ($^\circ$)	Std ($^\circ$)
$(0, 0)$	0.267	0.280
$(-2, +2)$	0.280	0.267
$(-5, +5)$	0.263	0.294
$(-10, +10)$	0.326	0.389

Table 2. Calibration results of rotational deviation within $(-5^\circ, +5^\circ)$ under different slight translations on TUMTraf-V2X dataset [15].

4.3. More comparative experimental results

We further evaluate our method against traditional LiDAR-camera calibration approaches [5] in the V2X scenario using the V2X-Seq dataset [11]. To ensure fairness, we conduct two experiments.

In the first, we manually set the initial rotation noise to match our method. As shown in Fig. 4, the result indicates that traditional methods offer little to no improvement in V2X settings. In the second, we follow the original setup using SuperGlue [7] to estimate initial extrinsics. The calibration result shows a large error of 126.0021°, suggesting that the method performs poorly.

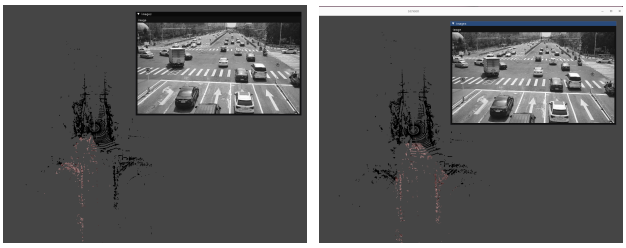


Figure 4. Calibration result (Left) of a traditional method [5]. Groud Truth (Right).

These results highlight the advantage of our method. This is mainly due to the sparse and repetitive scan pattern of spinning LiDARs, which limits reliable geometric and texture information from a single scan [5]. In V2X settings, vehicles are in motion and cannot accumulate multiple frames, making this calibration method ineffective.

4.4. Time and memory cost

Iteration is the key factor influencing the network's scale, runtime, and memory usage, with higher iterations generally improving accuracy (see Tab. 3). We performed ablation studies comparing our Mamba-based method and the Timesformer-based method at different iteration levels. Fig. 5 shows that as iterations increase, our method's memory inference cost becomes significantly lower than the Transformer-based approach, aligning with prior Mamba research. At low iterations, peak memory remains nearly unchanged since peak usage doesn't occur within the Mamba or Transformer modules during inference. Runtime increases similarly for both methods as iterations grow. Notably, the parameter reduction and accuracy gains over Timesformer are substantial and should not be overlooked. Overall, our method demonstrates clear advantages and strong potential.

4.5. Calibanyting on TUMTraf-V2X

Fig. 6 shows the result of our method (Left) and CalibAnything (Right) on the TUMTraf-V2X dataset [15], with initial error within 5°. Our method performs significantly better.

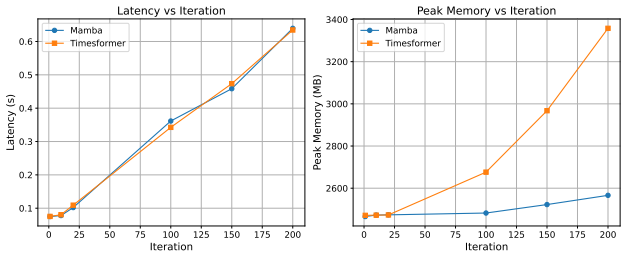


Figure 5. Time and peak memory cost during inference.

Iterations	Mean (°)	Std (°)
1	0.70	1.61
10	0.63	0.32
20	**0.61**	**0.21**

Table 3. Effect of iteration count on calibration accuracy.

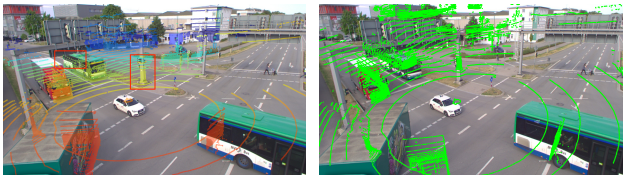


Figure 6. Calibanyting result on TUMTraf-V2X [15].

4.6. Patch-based or flatten

Tab. 4 compares three strategies for processing the calibration flow map. The results show that our method achieves a clear and significant advantage.

Methods	Mean (°)	Std (°)
patch+Mamba	**0.63**	**0.3**
directly project+Mamba	0.71	0.36
flatten+MLP	0.97	2.0

Table 4. Different calibration flow map processing strategies.

4.7. Qualitative results after calibration

Fig. 8 shows additional calibration results on the V2X-Seq dataset using MamV2XCalib. Our method can effectively address the issue of large deviations. Fig. 9 presents the visualization results where LCCNet [6] calibration fails dramatically, while our method succeeds. It is evident that our method demonstrates stronger robustness. In Fig. 10, we present additional calibration results on the TUMTraf-V2X dataset [15].

4.8. Results of the multi-range network iteration

Our method involves two iterative processes. On one hand, the calibration flow is iteratively updated through the GRU

module. On the other hand, during inference, we adopt the iterative alternation method of extrinsic parameter estimation and re-projection, consistent with previous works [6]. This process utilizes five models with identical structures trained under different ranges of deflection initializations. In Tab. 5, we provide detailed results after each network.

Network	Mean (°)				Std (°)			
	Total	Roll	Pitch	Yaw	Total	Roll	Pitch	Yaw
After ±20°	2.31	0.46	1.40	1.34	1.73	0.37	1.62	1.24
After ±10°	1.21	0.27	0.76	0.71	1.36	0.18	1.34	0.58
After ±5°	0.96	0.27	0.41	0.71	0.70	0.17	0.54	0.58
After ±2°	0.73	0.23	0.35	0.49	0.44	0.15	0.23	0.48
After ±1°	0.63	0.19	0.24	0.48	0.32	0.15	0.19	0.35

Table 5. The results of the multi-range network iteration.

4.9. Vehicle LiDAR vs Vehicle Camera

In implementing the V2X-based calibration strategy, we chose point clouds for vehicle-side data. Although point cloud data introduces challenges in cross-modal data processing, combining it with distance filtering methods can ensure a high degree of overlap between vehicle and roadside data. Fig. 7 compares the projections of all point clouds and those within the range of the vehicle's front-facing camera into the perception range of the roadside camera. It is clear that the 360-degree LiDAR helps achieve a greater data overlap compared to a single onboard camera.

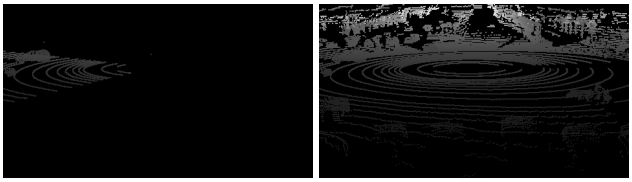


Figure 7. Projection (after 8×8 max pooling) of the point cloud within the forward-facing camera's range (left) and the entire point cloud in the roadside camera's viewpoint (right).

5. Generalization

Similar to other deep learning-based LiDAR calibration methods, our approach requires appropriate fine-tuning when extending to other datasets. Fortunately, roadside camera image data for calibration is easily accessible, and the usage scenarios for the same camera are relatively fixed. This facilitates fine-tuning on new datasets to achieve optimal calibration performance.

References

[1] Daniele Cattaneo and Abhinav Valada. Cmrnext: Camera to lidar matching in the wild for localization and extrinsic calibration. *IEEE Transactions on Robotics*, 2025. 2

[2] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024. 2

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 2

[4] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2

[5] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11301–11307. IEEE, 2023. 3

[6] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2894–2901, 2021. 2, 3, 4, 5

[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3

[8] Jieying Shi, Ziheng Zhu, Jianhua Zhang, Ruyu Liu, Zhenhua Wang, Shengyong Chen, and Honghai Liu. Calibrcnn: Calibrating camera and lidar by recurrent convolutional neural network and geometric constraints. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10197–10202. IEEE, 2020. 2

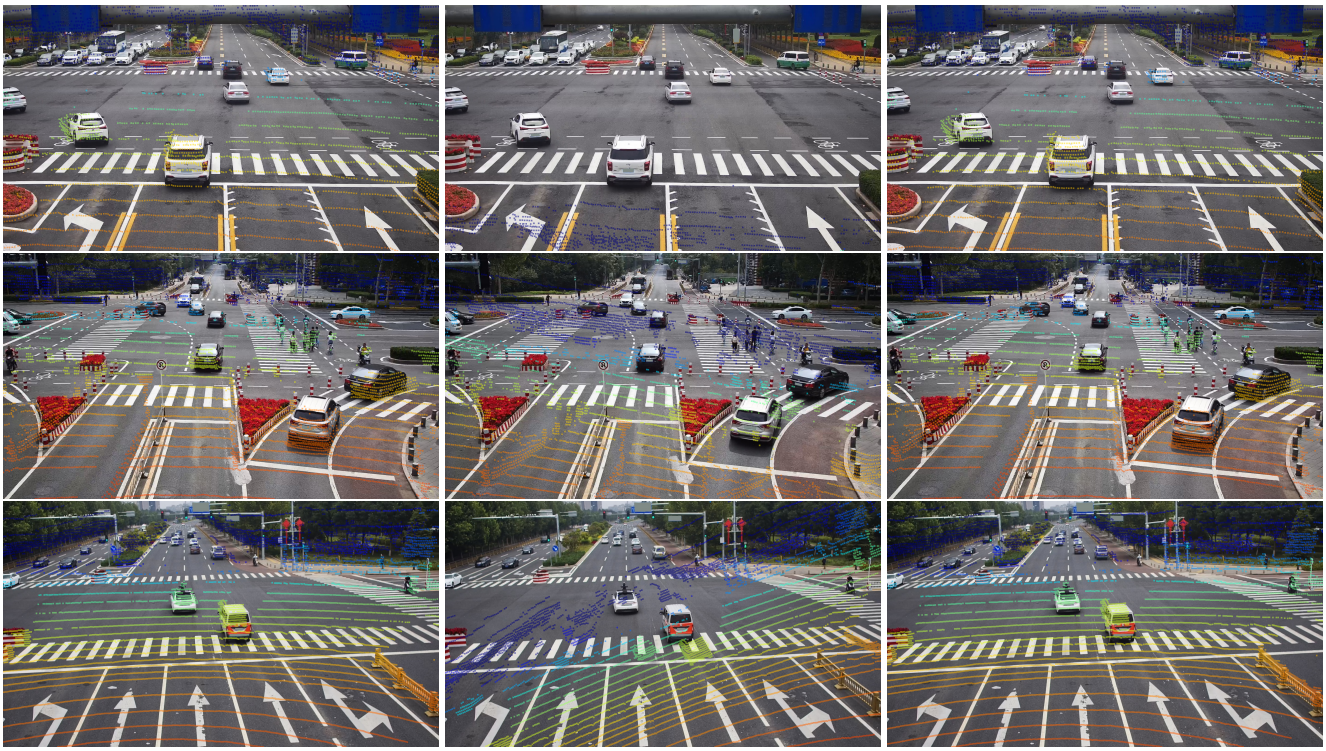
[9] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1

[10] Yuxuan Xiao, Yao Li, Chengzhen Meng, Xingchen Li, Jianmin Ji, and Yanyong Zhang. Calibformer: A transformer-based automatic lidar-camera calibration network. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16714–16720. IEEE, 2024. 2

[11] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, et al. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5486–5495, 2023. 1, 3

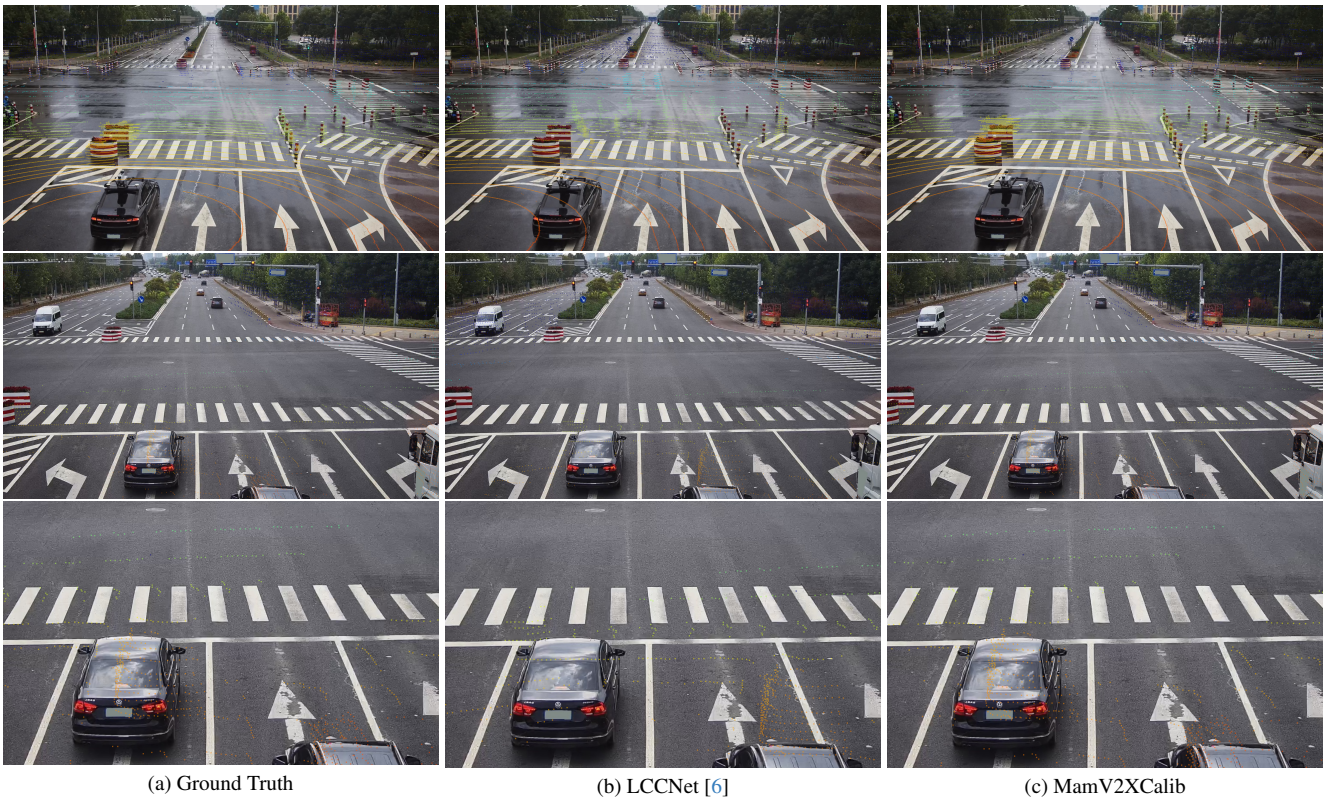
[12] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114:107861, 2021. 1

[13] Jiangtong Zhu, Jianru Xue, and Pu Zhang. Calibdepth: Unifying depth map representation for iterative lidar-camera online calibration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733. IEEE, 2023. 2



(a) Ground Truth (b) Before Calibration (c) MamV2XCalib

Figure 8. Qualitative LiDAR/Camera reprojection results on V2X-Seq dataset.



(a) Ground Truth (b) LCCNet [6] (c) MamV2XCalib

Figure 9. The failure of single-vehicle calibration methods [6] in V2X scenarios, while our method succeeds.

(a) Ground Truth (b) Before Calibration (c) MamV2XCalib

Figure 10. Qualitative LiDAR/Camera reprojection results on TUMTraf-V2X dataset.

[14] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2

[15] Walter Zimmer, Gerhard Arya Wardana, Suren Sritharan, Xingcheng Zhou, Rui Song, and Alois C Knoll. Tumtraf v2x cooperative perception dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22668–22677, 2024. 1, 2, 3