This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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

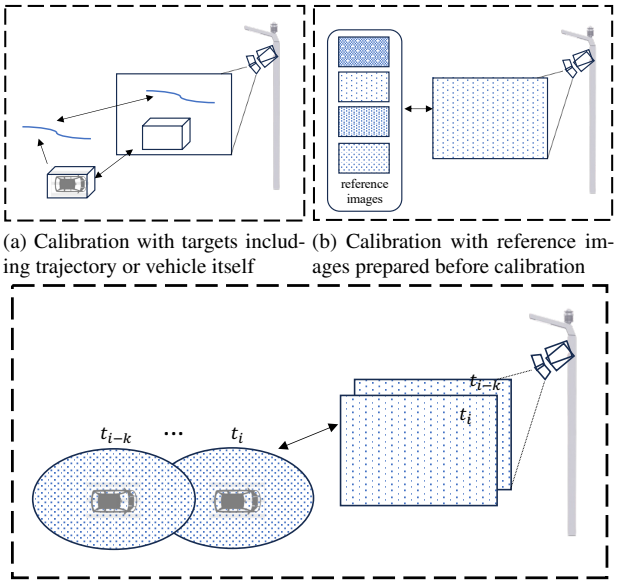
Yaoye Zhu Zhe Wang Yan Wang*
Institute for AI Industry Research (AIR), Tsinghua University
zhuyaoye22@gmail.com wangzhe@air.tsinghua.edu.cn wangyan@air.tsinghua.edu.cn

Abstract

As cooperative systems that leverage roadside cameras to assist autonomous vehicle perception become increasingly widespread, large-scale precise calibration of infrastructure cameras has become a critical issue. Traditional manual calibration methods are often time-consuming, labor-intensive, and may require road closures. This paper proposes MamV2XCalib, the first V2X-based infrastructure camera calibration method with the assistance of vehicle-side LiDAR. MamV2XCalib only requires autonomous vehicles equipped with LiDAR to drive near the cameras to be calibrated in the infrastructure, without the need for specific reference objects or manual intervention. We also introduce a new targetless LiDAR-camera calibration method, which combines multi-scale features and a 4D correlation volume to estimate the correlation between vehicle-side point clouds and roadside images. We model the temporal information and estimate the rotation angles with Mamba, effectively addressing calibration failures in V2X scenarios caused by defects in the vehicle-side data (such as occlusions) and large differences in viewpoint. We evaluate MamV2XCalib on the V2X-Seq and TUMTraf-V2X real-world datasets, demonstrating the effectiveness and robustness of our V2X-based automatic calibration approach. Compared to previous LiDAR-camera methods designed for calibration on one car, our approach achieves better and more stable calibration performance in V2X scenarios with fewer parameters. The code is available at https://github.com/zhuyaoye/MamV2XCalib.

1. Introduction

In recent years, due to the inherent limitations of on-board perception systems, many researchers have started focusing on utilizing external sensor information to assist autonomous vehicles [5, 10]. Among these external sources, infrastructure cameras are widely used in V2X systems for their wide deployment, broader field of view, and reduced occlusion [24]. However, outdoor deployment exposes roadside cameras to extreme weather and vibrations from large vehicles, causing shifts from their initial positions. If not recalibrated, such shifts can significantly degrade the accuracy of data fusion between road and vehicle sensors. In cases of large deviations, they may even result in a complete loss of monitoring over the intended area [38]. Therefore, efficient, scalable calibration methods are essential for stable intelligent transportation systems in outdoor environments.



(a) Calibration with targets including trajectory or vehicle itself (b) Calibration with reference images prepared before calibration

(c) Ours: V2X-based target-less camera calibration method

Figure 1. Comparison of roadside camera calibration methods.

Traditional calibration methods [32, 39] primarily rely on manual targets, but unlike vehicle sensors that can be calibrated indoors, fixed roadside cameras cannot be relocated. As the demand for camera calibration increases, these manual methods become time-consuming and labor-intensive and may cause disruptions to regular traffic operations. Re-

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

cent efforts focus on automated calibration [7, 30, 40]. Some methods use autonomous vehicle trajectories or vehicle detection targets. However, these approaches heavily depend on the accuracy of trajectory prediction and object detection, while overlooking rich background data. Other methods require the preparation of reference images or use real-world map images, employing virtual cameras aligned with these images as reference objects [6]. Preparing a series of reference images for each camera is both inconvenient and lacks real-time adaptability.

Whether traditional or modern methods, determining the position of a sensor in the world coordinate system requires two key conditions: known reference objects and the sensor's perception of these objects. In road scenarios, autonomous vehicles, being high-quality intelligent agents, possess real-time, robust capabilities to collect environmental data and accurately localize themselves. *Given this, can we invert the typical V2X process, using a vehicle's perception data to treat the entire environment as a reference, thereby calibrating multiple roadside cameras as the vehicle moves?*

We propose a novel V2X-based calibration algorithm that fuses vehicle-mounted LiDAR point clouds with roadside camera images. A neural network is then used to regress the camera's rotation deviation relative to its initial position. Unlike vehicle cameras, roadside cameras are typically hinge-mounted on infrastructure, limiting significant translational shifts. Minor deviations (centimeter-scale) have negligible impact on perception in V2X scenario [26, 35], whereas small rotational shifts, given the wider field of view, greatly degrade image quality (see Sec. 4.5). Thus, our primary focus is on the rotational deviation relative to the initial setup.

Inspired by the optical flow estimation [29, 33] and lidar-camera calibration task [17, 21], we project vehicle LiDAR point clouds onto roadside camera images to generate a depth map. Then, using a ResNet + FPN (Feature Pyramid Networks) [18], we extract multi-scale features to form a 4D correlation volume. We iteratively update a $2 \times H \times W$ feature matrix representing pixel-level correspondences between the depth map and the RGB image, using this matrix to associate new features from the 4D correlation volume. Our Mamba architecture further integrates spatial and temporal data, combining time-series information with feature matrices from prior iterations to overcome single-frame perception limitations. We divide the training process into two stages which will be explained in Sec. 4.2.

Compared to traditional roadside camera calibration methods, our approach fully utilizes environmental data, including but not limited to the vehicle itself. Meanwhile, it allows calibration at any road location as needed, without any pre-prepared reference equipment or images for these cameras, making it highly deployable. Compared to the existing LiDAR-camera calibration algorithms used on a single vehicle, our carefully designed network mitigates the issue of point clouds collected by vehicles in different frames occupying a relatively small and uncertain area in the roadside camera's view, delivering robust calibration results in V2X scenarios with fewer parameters.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose a target-less V2X-based roadside camera calibration solution.
- We designed a novel calibration network that combines multi-scale processing, iterative updates, and the mamba architecture to address the instability issues encountered when directly applying traditional LiDAR-camera calibration methods to V2X scenarios, while also improving calibration efficiency.
- We validated the effectiveness and robustness of our method on the V2X-Seq [37] and TUMTraf-V2X [43] datasets.

2. Related Works

Roadside camera calibration: A classic approach to roadside camera calibration is target-based calibration, typically using checkerboard patterns or other reference objects [39]. However, this method demands extensive manual effort and is impractical in busy traffic scenarios. To address this, some studies leverage vehicle-provided positioning or motion trajectories, aligning them with detection boxes [28, 30] or estimated trajectories [40] from roadside cameras. While eliminating the need for placing specific targets, these approaches depend heavily on detection accuracy, leading to error accumulation. Other methods exploit road geometry, such as vanishing points [12], or match perspective features like parallel curves [4], but suffer from limited accuracy in complex scenes. More recent work utilizes prepared reference images or 3D reconstructions of road environments [2, 6, 31], which are labor-intensive to collect and unsuitable for real-time deployment. CoCalibNet [36] introduces additional roadside sensors, but this increases roadside deployment costs and calibration workload. To overcome these limitations, we propose a fundamentally different, target-less calibration framework that utilizes vehicle sensor data to assist roadside camera calibration without relying on physical targets, vision-based detection, or pre-built scene representations.

LiDAR-camera calibration: LiDAR-camera calibration methods can be broadly categorized into target-based and target-less methods [17]. Traditional methods typically rely on specific calibration targets, such as checkerboards, spherical markers, or custom boards, transforming the calibration problem into an optimization problem. Recent research has shifted towards target-less methods, primarily applied to extrinsic parameter estimation for monoc-

ular camera and LiDAR systems on a single car. Li et al. [16] categorizes target-less calibration methods into four types: information-theoretic, feature-based, ego-motion-based, and learning-based approaches. Among these, deep learning methods have garnered widespread attention due to their promising results. RegNet [25] was the first to adopt a deep learning-based approach, using a network to extract features and then regress the parameters. CalibNet [11] introduced a geometry-supervised deep network to perform real-time calibration by maximizing geometric and photometric consistency. LCCNet [21] and CaLiCaNet [23] utilized cost volume to establish correspondences between features, which significantly improved upon previous deep learning-based calibration methods. Zhu et al. [41] use depth estimation to mitigate the modal differences. The latest researches [14, 34] attempt to enhance the form of cost volume using new architectures like Transformer. However, these deep learning-based methods perform poorly in V2X scenarios. That's because vehicle-mounted LiDAR often covers only a small portion, posing higher demands for cross-modal feature matching. Moreover, the covered area of depth maps remains nearly constant in the single vehicle's view, while the point clouds collected by the vehicle occupy different regions over time from a roadside perspective. Our method aims to address these problems by using temporal information and a better cost volume update mechanism.

State Space Models: Recently, State-Space Models (SSMs) have excelled in modeling long sequences, particularly in capturing long-range dependencies. The Structured State-Space Sequence (S4) [9] has been proposed as an alternative to Convolutional Neural Networks (CNNs) and Transformers, offering the advantage of linear complexity for efficient handling of long sequences. As research on S4 continues to deepen, various improved models have emerged, such as the S5 layer [27] which introduces MIMO SSMs and efficient parallel scanning and the Gated State-Space layer (GSS) [22]. These models have shown promising performance in language modeling tasks. Excitingly, the recent introduction of the data-dependent SSM layer has led to the development of Mamba [8], which outperforms Transformers of various sizes on large-scale real-world data and achieves linear scaling in sequence length. Recent studies have attempted to bring Mamba model into the visual domain. Vision Mamba [42] borrows the concept from Vision Transformers [19] by marking image sequences with position embeddings and compressing the visual representation using bidirectional state space models. Furthermore, Video Mamba [15] offers a linear-complexity approach for modeling dynamic spatiotemporal contexts, making it ideal for high-resolution long videos. Research indicates that the Mamba architecture has advantages over LSTM and Transformer architectures in video processing tasks. Building on Mamba, our work aims to achieve more efficient temporal modeling with reduced computational cost. We draw on visual-domain Mamba applications to integrate LiDAR and roadside camera temporal features at the representation level, thereby mitigating calibration failures caused by poor single-frame data quality.

3. Method

In this section, we present MamV2XCalib, a novel roadside camera calibration method. The framework of our approach is shown in Fig. 2. In the first stage, we review the mathematical definition of the roadside camera calibration problem (Sec. 3.1) so as to better understand our newly proposed calibration strategy. Next, we provide a detailed explanation of our network design, loss function, and inference process.

3.1. Problem Formulation and Input Processing

The modeling of a standard perspective camera can be described by the correspondence between 3D spatial points and pixels.

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

where $\begin{bmatrix} x_i & y_i & z_i & 1 \end{bmatrix}^T$ represents the homogeneous coordinates of a 3D point in an arbitrary known coordinate system A and $\begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$ denotes a homogeneous camera image coordinates of a picture. $\mathbf{K}$ denotes the camera's intrinsic parameters, while $\mathbf{R}$ and $\mathbf{t}$ represent the rotation and translation matrices, which transform the coordinates from system A to the camera coordinate system. d_i represents the projection depth of the 3D point. Specifically, in our method, A is set as the vehicle LiDAR coordinate system. Correspondingly, $\mathbf{R}$ and $\mathbf{t}$ represent the transformation from the onboard lidar coordinate system to the camera coordinate system. According to the description in the introduction, we assume that these parameters' initial values are given according to the expected calibration values, denoted as K_{init}, R_{init}, t_{init}, where K_{init} and t_{init} are stable and easy to deal with, and changes mainly occur in R_{init}:

$$T_{init} = \Delta T \cdot T_{LC} \tag{2}$$

$$\Delta T = \begin{bmatrix} \mathbf{R_{error}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3}$$

where T_{LC} represents the transformation from the current LiDAR coordinate system to the camera coordinate system. $\mathbf{R_{error}}$ is the rotational deviation. T_{init} is artificially set and our goal is to automatically obtain the value of ΔT through sensor data, which can determine the current position of the

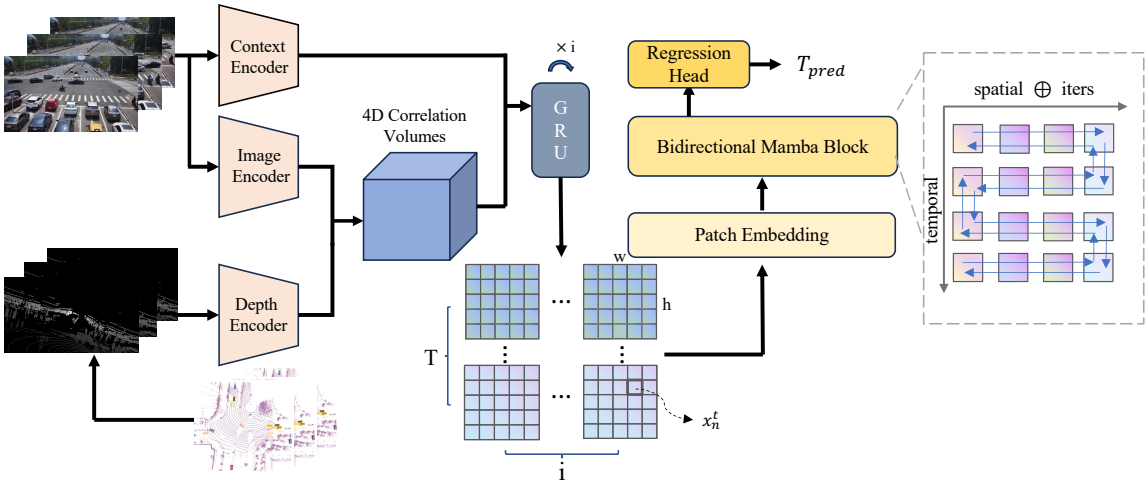


Figure 2. The overall framework of MamV2XCalib. Initially, the point clouds are projected into the camera view using inaccurate initial extrinsic parameters. The network input consists of multiple frames of erroneous depth maps and roadside camera images. Multi-scale features are extracted from the images and depth maps through three channels, forming 4D correlation volumes. A GRU block is utilized to recurrently update the pixel-level correspondences between the images and depth maps (calibration flow) by retrieving features from the volume based on existing estimates. A series of calibration flows generated from multiple frames and iterations is divided into patches and fed into the Mamba model, with temporal and spatial embeddings. An additional token summarizes the features to regress the rotation deviation value.

roadside camera in the world coordinate system. Before being fed into the network, the sensor data needs to be aligned to a unified coordinate system. The vehicle's point cloud is projected onto the roadside camera plane using the initial parameters T_{init} and Eq. 1.

3.2. Multi-scale feature extraction

Our method employs three feature extraction branches to separately extract image information, depth map information, and image context. The two branches processing camera images both use a pre-trained ResNet-18 as the backbone. To obtain richer multi-scale features, and considering that highly abstract feature maps are not conducive to the subsequent search for pixel-level relationships, we fuse the upsampled feature map of latter layers with the former layers to obtain the final feature map. Finally, the output of image encoder F_1 and $F_{context}$ is $\mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times D}$ where we set $D = 256$. The depth map branch architecture F' is similar to the image branch, except that the input channel is modified to one dimension, and pre-trained parameters are not loaded. Due to the unique nature of depth maps, the three branches do not share parameters.

3.3. Feature Matching and Iterative refinement

We directly associate the correspondences between roadside images $I_1 \in \mathcal{I}$ and depth maps $I_2 \in \mathcal{D}$ with the deviation of the roadside camera's extrinsic parameters. This pixel-level correspondence can be regarded as a special type of flow field, which we refer to as the *calibration flow* in this paper. Inspired by the RAFT [29] model used for optical flow estimation, we construct a multi-scale 4D correlation volume as follows:

$$Inn(F_1(I_1), F'(I_2)) \in \mathbb{R}^{h\times w\times h\times w} \tag{4}$$

$$U_k = \text{Pool}(Inn) \in \mathbb{R}^{h\times w\times \frac{h}{2^k}\times \frac{w}{2^k}} \tag{5}$$

where Inn is calculated through inner product and Pool is designed to get multi-scale 4D correlation volume. Through these operations, the similarity between any two pixels is stored in this volume. Next, we repeatedly query features from the 4D dictionary to estimate pixel correspondences based on the former correspondences. We initialize the flow field (pixel correspondences) to zeros. In each iteration, given the feature flow map, we take each pixel's corresponding point as the center and get the similarity of all pixels within a distance of r. We call this operation LookUp.

$$h'' = \text{GRU}(\text{LookUp}\{U_k, f, r\}, h', F_{context}(I_1)) \tag{6}$$

$$df = \text{Flow}(h'') \tag{7}$$

We then input the features we just looked up, hidden states from last iteration, and context features together into the GRU module, obtaining new hidden states. The flow residuals df is predicted by layer Flow consisting of two convolutional layers. The residual df is added to the previous

flow field to update it. In this way, we obtain a series of flow field estimates $f_i \in \mathcal{F}$ equal to the number of iterations.

To ensure that the model can get calibration flow fields, in the first stage of training, we directly regress the extrinsic parameters from each flow field obtained at every step using two fully connected layers.

3.4. Global Aggregation with Mamba

While our 4D correlation volume effectively captures global correlations between inaccurate depth maps and images, the unique V2X scenario poses additional challenges. The already sparse point clouds become even sparser due to viewpoint disparities between roadside and vehicle perspectives. Moreover, the effective regions of depth maps shift over time, making it inherently difficult to derive accurate calibration flow fields in uncertain, data-scarce regions using single-frame data alone. Directly regressing extrinsics from all correspondences would inevitably introduce bias.

Fortunately, although the vehicle is in motion, the roadside camera's deviation from its initial position remains fixed over a calibration period. This stability allows us to perform temporal fusion, integrating data across frames to overcome the limitations described above. We select Mamba because our sequence of calibration flow fields has dimensions akin to video data, where Transformers excel in fusion tasks. However, our iterative process produces a large number of tokens, prompting us to adopt the latest Mamba [15, 42] for its superior efficiency in handling such data.

Specifically, we extend the input from a single frame to a sequence. According to the method in Sec. 3.3, we can obtain a calibration flow field of size $\mathbb{R}^{T\times i\times h\times w}$. We treat the iterative process and temporal dimension as two separate axes. First, we concatenate the flow field maps obtained from each frame into an overall map of size $\mathbb{R}^{T\times h\times (w\times i)}$. Then we divide it into a series of patches x_n^t instead of flattening them simply and project them into a hidden space through convolutional layers. The patches of the frames are arranged as shown in Fig. 2. Similar to [1, 15], we add positional and temporal embeddings p_s and p_t to retain the spatiotemporal distinctions and insert an additional token z_{add} into the entire sequence to summarize the information.

$$Z = [z_{add}, \mathbf{E}x_1^1, \mathbf{E}x_2^1, \ldots, \mathbf{E}x_N^T] + p_s + p_t \tag{8}$$

$$z_{add} = \mathbf{Mam}(Z) \tag{9}$$

where the projection by $\mathbf{E}$ is equivalent to 3D convolution and $\mathbf{Mam}$ is the bidirectional mamba block. Through the Mamba model, we construct state space and scan it as illustrated, enabling the model to selectively model different regions of each flow field map from a spatial perspective. Additionally, this allows the model to automatically associate features generated from data with varying quality at different timestamps. The added token summarizes the series of lengthy calibration flow maps generated through iterations and temporal processes.

After obtaining the refined summarized features z_{add}, we use two fully connected layers to regress the quaternion, achieving the desired estimation of the rotational error.

3.5. Loss Function

For each obtained rotation prediction, our loss has the following form:

$$L_i = \lambda_r L_r + \lambda_p L_p \tag{10}$$

where λ_r and λ_p denote respective loss weights. L_r is the rotation loss and we use angular distance to represent the difference between quaternions. Considering that our method utilizes coordinate transformations of point clouds P_i, we calculate the distance between the two point clouds after the actual transformation and the predicted transformation to obtain L_p:

$$L_p = \frac{1}{N}\sum_{i=1}^{N}\left\| T_{LC}^{-1}\cdot T_{pred}^{-1}\cdot T_{init}\cdot P_i - P_i \right\|_2 \tag{11}$$

where T_{LC} is the ground truth LiDAR-Camera extrinsic matrix and T_{pred} is the output of the calibration network.

In the first training stage, we iteratively obtain n estimates per frame, with the loss as a weighted sum of their individual losses using exponentially increasing factors.

$$L = \sum_{i=1}^{n}\gamma^{n-i}L_i \tag{12}$$

In the second stage, multi-frame data is input, with each frame producing a single estimate, and the loss is defined as in Eq. 10.

3.6. Calibration Inference

One of our inputs, the depth map, is influenced not only by the quality of the point cloud itself but also by the initially given extrinsic parameters. To address significant deviations and maintain consistency with LCCNet [21], we concatenate multiple networks that have the same architecture but are trained under different noise conditions. We then reproject the depth maps based on the predictions T_i from the previous network to obtain higher-quality input data. Therefore, during the final inference, we have:

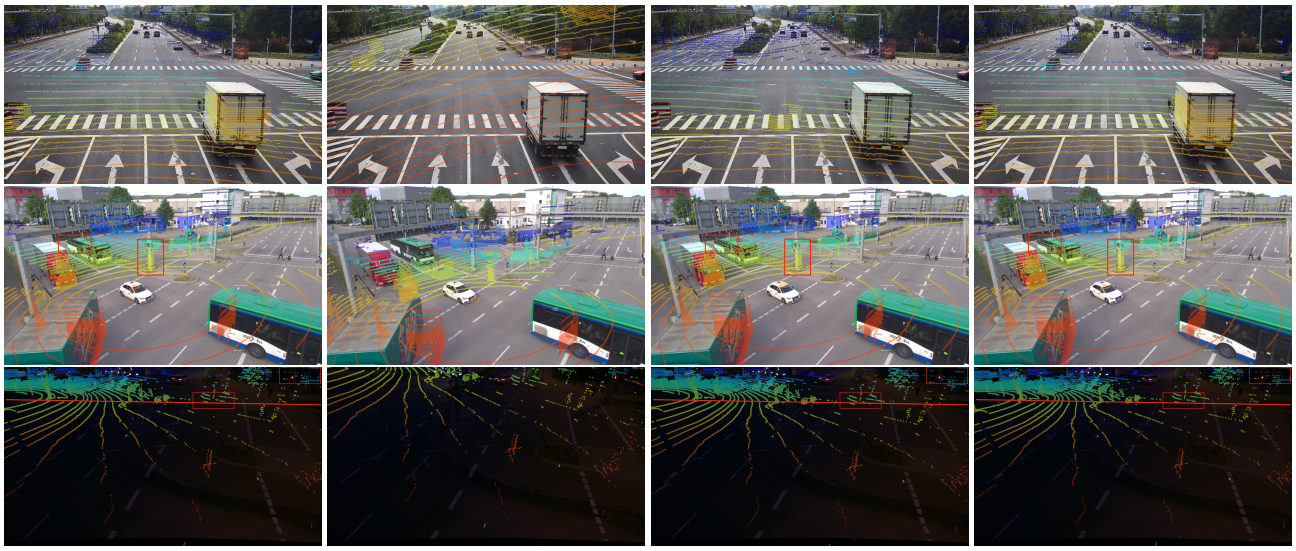
$$\tilde{T}_{LC} = (T_0 \cdot T_1 \ldots T_n)^{-1} T_{init} \tag{13}$$

4. Experiment

We conduct extensive experiments on two real-world datasets, V2X-Seq [37] and TUMTraf-V2X [43]. In this section, we will specifically introduce the training details and present the experimental results.

Deviation	Method	Mean (°)				Std (°)			
		Total	Roll	Pitch	Yaw	Total	Roll	Pitch	Yaw
$(-20°, +20°)$	LCCNet [21]	1.0741	**0.1788**	0.6056	0.6455	2.5336	0.1537	2.4624	0.7920
$(-20°, +20°)$	Calib-anything [20]	-	-	-	-	-	-	-	-
$(-20°, +20°)$	Ours	**0.6313**	0.1867	**0.2409**	**0.4780**	**0.3211**	**0.1489**	**0.1860**	**0.3500**
$(-10°, +10°)$	LCCNet [21]	0.7909	**0.1792**	0.3440	0.5548	1.6825	**0.1111**	1.6147	0.5985
$(-10°, +10°)$	Calib-anything [20]	-	-	-	-	-	-	-	-
$(-10°, +10°)$	Ours	**0.5858**	0.2260	**0.2441**	**0.4001**	**0.2428**	0.1525	**0.1745**	**0.2776**

Table 1. Comparison results with other calibration methods on V2X-Seq dataset [37]. Calib-anything [20] fails. (See Fig. 4)



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

Figure 3. Qualitative reprojection results. First row: On V2X-Seq dataset, within the initial large deflection range of $(-20°, +20°)$, LCCNet [21] shows significant deviation in some scenes, while our method remains accurate. Second row: TUMTraf-V2X (daytime). Third row: TUMTraf-V2X (nighttime). The initial deflections are within a small range of $(-5°, +5°)$ and our method outperforms better.

4.1. Dataset

V2X-Seq: Yu et al. [37] introduce a large-scale real-world cooperative vehicle-infrastructure dataset. It contains a large amount of time-synchronized data from vehicle-side and roadside sensors, as well as transformations between multiple coordinate systems.

TUMTraf-V2X: There is another V2X dataset named TUMTraf-V2X [43]. It focuses on challenging traffic scenarios and various day and nighttime scenes. We utilize vehicle-side LiDAR data and roadside camera data.

Similar to single-vehicle LiDAR-camera calibration tasks, we add random noise within a specified range to the extrinsic parameters to generate abundant training data. Specifically, after applying a random perturbation ΔT to T_{LC}, we obtain the initial extrinsic parameter $T_{init} = \Delta T \cdot T_{LC}$. By randomly changing the deviation value, we can acquire a large amount of training data.

4.2. Training Details

We train our network on a single A100 GPU, with the training process divided into two stages. Next, I will elaborate on the training process and use the hyperparameter settings on V2X-Seq [37] as an example.

In the first stage, we perform calibration using only single-frame data. Specifically, we exclude the Mamba module and directly regress the extrinsic parameters from the calibration flow map. This approach allows the network to focus on accurately predicting the calibration flow map. We set the initial learning rate to 0.00003 and train for 70 epochs, after which the learning rate is reduced until convergence. All available paired data is utilized in this stage.

In the second stage, we train the Mamba module based on the first stage, enabling the network to fuse multiple calibration flow maps generated over time. The initial learning rate is set to 0.0001 and decays during training. We begin

Experiment	Temporal	Iteration	4D Correlation Volume	Mean (○)	Std (○)	Param (M)
Ours	Mamba[8]	✓	✓	0.6313	0.3211	47
w/o Mamba (+Trans)	Timesformer[3]	✓	✓	0.6683	0.3808	55
w/o Temporal	-	✓	✓	0.9742	2.0013	41
w/o Iteration	Mamba[8]	-	✓	0.7036	1.6162	47
w/o Temporal and Iteration	-	-	✓	1.1025	2.9178	47
3D feature matching [21]	-	-	-	1.0741	2.5336	70

Table 2. Ablation study on the effect of different components of our model.

by freezing the network components preceding the Mamba module and then jointly update all network parameters. To enhance training data quality, we use data where the horizontal distance between vehicles and infrastructure is less than 50 meters.

4.3. Evaluation

During evaluation, we performed distance filtering before inputting the data into the network. The calibration system is activated only when the vehicle is within certain distance. We will explain the benefits of this operation in Sec. 4.5.

On V2X-Seq Dataset [37]: Tab. 1 highlights the effectiveness of our method in roadside camera calibration tasks. For a mis-calibration range of $(-20°, +20°)$, our approach achieves a mean calibration error of $0.6313°$, with mean Euler angle errors of $[0.1867°, 0.2409°, 0.4780°]$. We compare our method to a high-performing single-vehicle LiDAR-camera calibration technique [21], demonstrating that our approach offers superior accuracy in V2X scenarios over directly applying the single-vehicle method. Furthermore, we assess its robustness. Our method significantly reduces the likelihood of extreme calibration failures, lowering the standard deviation of the angle-error from $2.5336°$ to $0.3211°$—a reduction by an order of magnitude. The deviations within this range are manageable by existing perception methods [35]. When the initial error is reduced to $(-10°, +10°)$, our method yields even better results while maintaining greater robustness and a lower risk of severe errors. Additionally, we present a qualitative comparison with the latest train-free LiDAR-camera calibration method [20] in Fig. 4. In V2X scenarios with substantial camera deflection, the competing method fails, whereas our approach consistently succeeds. We also compared with a traditional LiDAR-camera calibration method [13] (see Appendix 4.3 for details).

On TUMTraf-V2X Dataset [43]: Tab. 3 presents comparative results on the TUMTraf-V2X dataset, where our method remains competitive. For a mis-calibration range of $(-5°, +5°)$, our approach achieves a mean calibration error of $0.267°$. Remarkably, our calibration strategy proves effective even in nighttime scenarios (see Fig. 3).

To ensure a fair comparison, we retrained LCCNet [21] on the same dataset and modified its output to predict only rotational errors, aligning it with our network's configuration. Fig. 3 shows that after passing through our calibration network, effective calibration can still be achieved despite significant initial extrinsic parameter deviations. Based on the calibrated position of the roadside camera, the vehicle point clouds can be well-matched with the images from the perspective of the roadside camera.



Figure 4. Ground Truth (Left). Calib-Anything [20] (Middle) fails to calibrate within (-20°, +20°), while ours (Right) succeeds.

Deviation	Method	Mean (°)	Std (°)
$(-5°, +5°)$	LCCNet [21]	0.389	0.608
$(-5°, +5°)$	Ours	**0.267**	**0.280**

Table 3. Comparison results on TUMTraf-V2X dataset [43].

4.4. Ablations and Analysis

We conducted a series of ablation experiments to demonstrate the importance of each component.

Mamba: First, we removed the Mamba component used for spatiotemporal fusion while retaining feature flow estimation for direct regression. Results are shown in Tab. 2. It can be observed that the Mamba module significantly improves calibration stability, greatly reducing the standard deviation of the estimated values, which indicates the necessity of this module. Additionally, we attempted to replace it with a Transformer-based Timesformer [3] module. It can be seen that the Mamba-based method achieves relatively better results with a smaller number of parameters.

Iteration: Our method refines the calibration flow progressively through iteration. During inference, we iterate 10 times to generate 10 calibration flow maps, which together provide spatial dimensional information. We verified

that using only a single iteration to produce the calibration flow during inference significantly weakens the network's capability, indicating that generating calibration flow maps through multiple iterations is essential.

Multi-scale 4D Correlation Volume: We replace the 4D correlation volume with the 3D feature matching layer in LCCNet [21], and find it costs much more parameters and perform worse than 4D correlation volume with iteration (marked by w/o Temporal).

4.5. Other experiments about V2X Calibration

Why Ignore Translation Noise: Fig. 5 shows the point cloud projection deviations caused by translational and rotational errors in a V2X scenario. It is evident that minor translational have a negligible impact on the perception system and can be disregarded. In contrast, rotational errors often lead to significant matching discrepancies in vehicle-road data. The experiment in the appendix demonstrate that such slight deviations do not significantly affect rotational calibration. We attribute this to the fact that, in V2X scenarios, the absolute distance between the vehicle and the road typically far exceeds the potential translational offset of roadside cameras due to environmental factors. Therefore, it is reasonable to simplify the calibration problem of roadside cameras by focusing on rotation.

Necessity of the Mamba Module: To further substantiate the importance of the Mamba module, we expanded our ablation study by comparing pre-fusion and post-fusion approaches. As shown in Tab. 4, on the V2X-Seq dataset [37], our method markedly outperforms two alternatives: directly stacking multi-frame point clouds or applying post-processing to the frames using a single-frame technique.

Distance Filtering: By utilizing vehicle mobility, we can easily choose data with shorter distances between vehicles and infrastructure for calibration. This distance can be calculated before feeding it into the neural network, allowing us to initially evaluate data quality based on distance to boost calibration accuracy. This is an advantage that earlier methods lack. In Fig. 6, we show the mean error and standard deviation of calibration using vehicle point clouds across various distance ranges. Experiments demonstrate that filtering data by distance improves data quality and strengthens the method's robustness.

Migration to single car calibration: Although our method was originally designed for V2X, it can naturally be extended to single car where translation errors must be considered. The results are shown in the attachment.

4.6. Limitation

Our method relies on vehicle-side LiDAR perception data, which places certain demands on the quality of vehicle-side sensors. Additionally, although the V2X-based calibration strategy offers benefits such as target-free operation, high flexibility, and large-scale deployment, it also introduces vehicle-to-infrastructure information interaction, which imposes higher requirements on time synchronization. Besides, similar to other deep learning-based calibration methods, fine-tuning is necessary to achieve good results when transferring to entirely different datasets.



Figure 5. The impact of extrinsic noise. Ground Truth (Left). Translated within 10 cm (Middle). Rotated within 20° (Right).

Fusion Methods	Mean (°)	Std (°)
stacking points	0.952	2.184
average	0.974	2.001
least square fitting	0.820	1.254
ours	**0.631**	**0.321**

Table 4. Comparison with other fusion methods on the V2X-Seq dataset [37] (initial deviation $(-20^{\circ}, +20^{\circ})$).

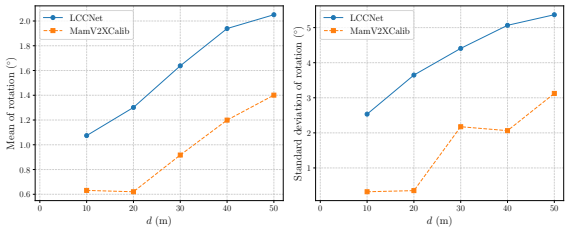


Figure 6. Calibration result on V2X-Seq dataset [37] after filtering vehicle point cloud data with a horizontal distance between the vehicle and roadside camera of distance $< d$.

5. Conclusion

This paper proposes a target-less infrastructure camera calibration strategy based on vehicle-road collaboration and introduces a novel method designed specifically for V2X scenarios. Compare to existing approaches, our strategy makes full use of environmental information and is suited for busy road scenarios. We present an iterative method that establishes pixel-level correspondences between depth maps and images. By incorporating the Mamba model's strengths in temporal modeling, we overcome instability challenges in single-vehicle calibration applied to V2X contexts with fewer parameters. Our method proves effective and robust on two datasets. We hope that it can inspire future related study.

Acknowledgements

This work is supported by National Science and Technology Major Project (2022ZD0115502), and Wuxi Research Institute of Applied Technologies, Tsinghua University under Grant 20242001120.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 5

[2] Esra Ataer-Cansizoglu, Yuichi Taguchi, Srikumar Ramalingam, and Yohei Miki. Calibration of non-overlapping cameras using an external slam system. In *2014 2nd International Conference on 3D Vision*, pages 509–516. IEEE, 2014. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 7

[4] Eduardo R Corral-Soto and James H Elder. Automatic single-view calibration and rectification from parallel planar curves. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 813–827. Springer, 2014. 2

[5] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17252–17262, 2022. 1

[6] Giacomo D'Amicantonio, Egor Bondarev, et al. Automated camera calibration via homography estimation with gnns. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5876–5883, 2024. 2

[7] Markéta Dubská, Adam Herout, Roman Juránek, and Jakub Sochor. Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1162–1171, 2014. 2

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3, 7

[9] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 3

[10] Lei Huang and Wenzhun Huang. Rd-yolo: An effective and efficient object detector for roadside perception system. *Sensors*, 22(21):8097, 2022. 1

[11] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1110–1117. IEEE, 2018. 3

[12] Viktor Kocur and Milan Ftáčnik. Traffic camera calibration via vehicle vanishing point detection. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V 30*, pages 628–639. Springer, 2021. 2

[13] 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. 7

[14] Yu-Chen Lee and Kuan-Wen Chen. Lccraft: Lidar and camera calibration using recurrent all-pairs field transforms without precise initial guess. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16669–16675. IEEE, 2024. 3

[15] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pages 237–255. Springer, 2025. 3, 5

[16] Xingchen Li, Yuxuan Xiao, Beibei Wang, Haojie Ren, Yanyong Zhang, and Jianmin Ji. Automatic targetless lidar–camera calibration: a survey. *Artificial Intelligence Review*, 56(9):9949–9987, 2023. 3

[17] Kang Liao, Lang Nie, Shujuan Huang, Chunyu Lin, Jing Zhang, Yao Zhao, Moncef Gabbouj, and Dacheng Tao. Deep learning for camera calibration and beyond: A survey. *arXiv preprint arXiv:2303.10559*, 2023. 2

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[20] Zhaotong Luo, Guohang Yan, and Yikang Li. Calib-anything: Zero-training lidar-camera extrinsic calibration method using segment anything. *arXiv preprint arXiv:2306.02656*, 2023. 6, 7

[21] 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, 5, 6, 7, 8

[22] Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022. 3

[23] Arya Rachman, Jürgen Seiler, and André Kaup. End-to-end lidar-camera self-calibration for autonomous vehicles. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023. 3

[24] Bernhard Rinner and Wayne Wolf. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10): 1565–1575, 2008. 1

[25] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE, 2017. 3

[26] Christoph Schöller, Maximilian Schnettler, Annkathrin Krämmer, Gereon Hinz, Maida Bakovic, Müge Güzet, and Alois Knoll. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3934–3941. IEEE, 2019. 2

[27] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 3

[28] Jakub Sochor, Roman Juránek, and Adam Herout. Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*, 161:87–98, 2017. 2

[29] 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. 2, 4

[30] Alexander Tsaregorodtsev, Adrian Holzbock, Jan Strohbeck, Michael Buchholz, and Vasileios Belagiannis. Automated static camera calibration with intelligent vehicles. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2023. 2

[31] Khiem Vuong, Robert Tamburo, and Srinivasa G Narasimhan. Toward planet-wide traffic camera calibration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8553–8562, 2024. 2

[32] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei-Yue Wang. Research on lane-marking line based camera calibration. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6. IEEE, 2007. 1

[33] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2025. 2

[34] 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. 3

[35] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen. Bevheight: A robust framework for vision-based roadside 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21611–21620, 2023. 2, 7

[36] Chen Yaqing and Wang Huaming. Robust extrinsic calibration for lidar-camera systems via depth and height complementary supervision network. *IEEE Access*, 2025. 2

[37] 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. 2, 5, 6, 7, 8

[38] Xinyu Zhang, Yijin Xiong, Qianxin Qu, Renjie Wang, Xin Gao, Jing Liu, Shichun Guo, and Jun Li. Cooperative visual-lidar extrinsic calibration technology for intersection vehicle-infrastructure: A review. *arXiv preprint arXiv:2405.10132*, 2024. 1

[39] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 1, 2

[40] Cong Zhao, Delong Ding, Yupeng Shi, Yuxiong Ji, and Yuchuan Du. Graph matching-based spatiotemporal calibration of roadside sensors in cooperative vehicle-infrastructure systems. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 2

[41] 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. 3

[42] 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. 3, 5

[43] 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. 2, 5, 6, 7