TUMTraf V2X Cooperative Perception Dataset

Walter Zimmer[1] Gerhard Arya Wardana[1] Suren Sritharan[1]
Xingcheng Zhou[1] Rui Song[1,2] Alois C. Knoll[1]

[1]Technical University of Munich [2]Fraunhofer IVI

https://tum-traffic-dataset.github.io/tumtraf-v2x

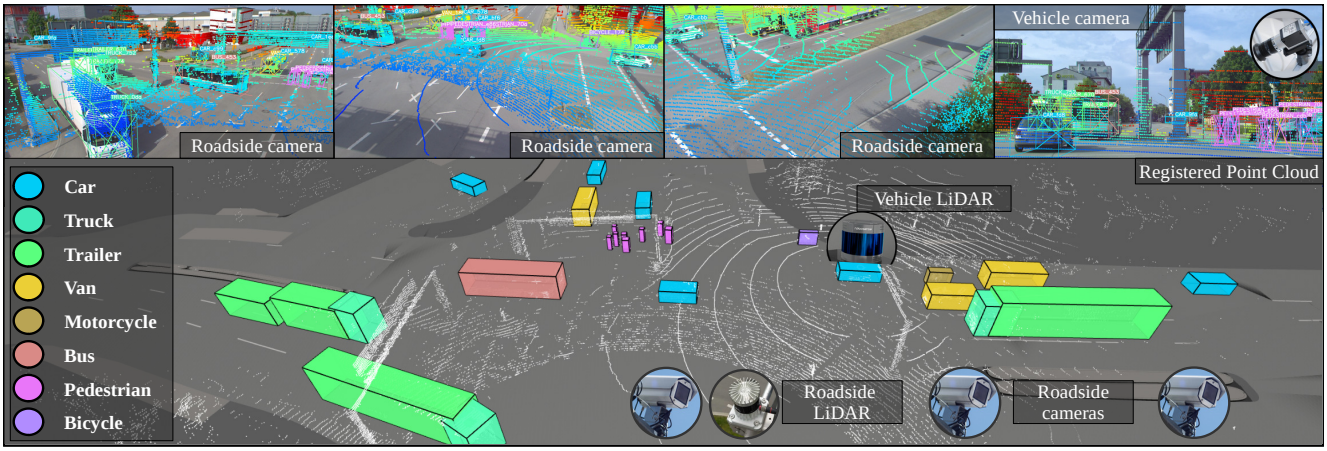


Figure 1. **Visualization** of 3D box labels and tracks in our **TUMTraf V2X Cooperative Perception Dataset**. The top part shows the labels projected into the four camera images. The part below shows a point cloud from two LiDARs with 3D box labels of the same scene.

Abstract

Cooperative perception offers several benefits for enhancing the capabilities of autonomous vehicles and improving road safety. Using roadside sensors in addition to onboard sensors increases reliability and extends the sensor range. External sensors offer higher situational awareness for automated vehicles and prevent occlusions. We propose CoopDet3D, a cooperative multi-modal fusion model, and TUMTraf-V2X, a perception dataset, for the cooperative 3D object detection and tracking task. Our dataset contains 2,000 labeled point clouds and 5,000 labeled images from five roadside and four onboard sensors. It includes 30k 3D boxes with track IDs and precise GPS and IMU data. We labeled eight categories and covered occlusion scenarios with challenging driving maneuvers, like traffic violations, near-miss events, overtaking, and U-turns. Through multiple experiments, we show that our CoopDet3D camera-LiDAR fusion model achieves an increase of +14.36 3D mAP compared to a vehicle camera-LiDAR fusion model. Finally, we make our dataset, model, labeling tool, and dev-kit publicly available on our website.

1 . Introduction

Cooperative perception involves the fusion of onboard sensor data and roadside sensor data, and it offers several advantages for enhancing the capabilities of autonomous vehicles and improving road safety. Using data from multiple sources makes the perception more robust to sensor failures or adverse environmental conditions. Roadside sensors provide an elevated view that helps to detect obstacles early. Moreover, they are also beneficial for precise vehicle localization and reduce the computational load of automated vehicles by offloading some perception tasks to the roadside sensors. Roadside sensors provide a global perspective of the traffic and offer a comprehensive situational awareness when fused with onboard sensor data. There are also fewer false positives or negatives because cooperative perception cross-validates the information from different sensors.

Infrastructure sensors can share perception-related information with vehicles through V2X. Due to minimal delay, and real-time capabilities, the infrastructure-based perception systems can further enhance the situational awareness and decision-making processes of vehicles.

Intelligent Transportation Systems (ITS) like the Testbed

Table 1. Comparison of 3D cooperative V2X perception datasets with our proposed TUMTraf-V2X Cooperative Perception dataset (I=Infrastructure, V=Vehicle).

Dataset	OPV2V [51]	V2XSet [49]	V2X-Sim [30]	V2V4Real [52]	DAIR-V2X-C [57]	V2X-Seq (SPD) [59]	**TUMTraf-V2X (Ours)**
Year	2022	2022	2022	2022	2022	2023	2024
V2X	V2V	V2V&I	V2V&I	V2V	V2I	V2I	V2I
Real data	-	-	-	✓	✓	✓	✓
Annotation range	120 m	120 m	70 m	200 m	280 m	280 m	200 m
Day & night scenes	-	-	-	-	✓	✓	✓
# object classes	1	1	1	5	10	9	8
Track IDs	-	-	✓	✓	-	✓	✓
HD Maps	✓	✓	✓	✓	-	✓	✓
# of sensors (I | V)	- | 6*	- | 6*	5 | 7	- | 8‡	2 | 3	2 | 3	5 | 4
Available worldwide	✓	✓	✓	✓	-	-	✓
Traffic violations	-	-	-	-	-	-	✓
Labeled attributes#	-	-	-	-	-	-	✓
OpenLABEL format	-	-	-	-	-	-	✓
# Point Clouds	11k	11k	10k	20k	39k	15k	2.0k
# Images	44k	44k	60k	40k†	39k	15k	5.0k
# 3D Boxes	233k	233k	26k	240k	464k	10.45k	29.38k
Location	CARLA	CARLA	CARLA	USA	China	China	Germany

† Image dataset has not been released yet.
* Value per vehicle. Multiple Conn. and Autom. Vehicles (CAVs) are used.
‡ Total sensors from 2 CAVs.
# Weather, time of day, orientation, number of LiDAR points

for Autonomous Driving [25] aim to improve safety by providing real-time traffic information. According to [14], testbeds extensively start using LiDAR sensors in their setups to create an accurate live digital twin of the traffic. Connected vehicles get a far-reaching view which enables them to react to breakdowns or accidents early. ITS systems also provide lane and speed recommendations to improve the traffic flow.

The key challenge with ego-centric vehicle datasets is that there are many occlusions from a vehicle perspective, e.g., if a large truck in front of the ego vehicle obscures the view. Roadside sensors located at a smart intersection provide a broad overview of the intersection and a full-surround view. Given the immense potential of ITS, there is a specific need for V2X datasets. Despite the high costs associated with collecting and labeling such datasets, this work addresses this challenge as a crucial step toward realizing large-scale ITS implementations.

Our contributions are as follows:

- We provide a high-quality V2X dataset for the cooperative 3D object detection and tracking task with 2,000 labeled point clouds and 5,000 labeled images. In total, 30k 3D bounding boxes with track IDs were labeled in challenging traffic scenarios like near-miss events, overtaking scenarios, U-turn maneuvers, and traffic violation events.
- We open-source our 3D bounding box annotation tool (*3D BAT* v24.3.2) to label multi-modal V2X datasets.
- We propose *CoopDet3D*, a cooperative 3D object detection model, and show in extensive experiments and ablation studies that it outperforms single view models on our V2X dataset by +14.3 3D mAP.
- Finally, we provide a development kit to load the annotations in the widely recognized and standard format OpenLABEL [20], to facilitate a seamless integration and utilization of the dataset. Furthermore, it can preprocess, visualize, and convert labels to and from different dataset formats, and evaluate perception and tracking methods.

2 . Related work

3D autonomous driving datasets are mainly categorized based on the viewpoint. Table 1 highlights the main differences between our proposed dataset and other V2X datasets.

2.1. Single viewpoint datasets

Single viewpoint datasets are obtained from a single point of reference, either an ego-vehicle or roadside infrastructure. Onboard sensor-based datasets like KITTI [19], nuScenes [9], and Waymo [42] contain a diverse set of sensor data collected from a moving vehicle equipped with multiple sensors, including high-resolution cameras, LiDARs, radars, and GPS/INS systems. These datasets are abundant and provide many annotated data, including bounding boxes, track IDs, segmentation masks, and depth maps under different urban driving scenarios.

On the other hand, roadside sensor-based datasets are in the infancy stage. High-quality multi-modal (camera

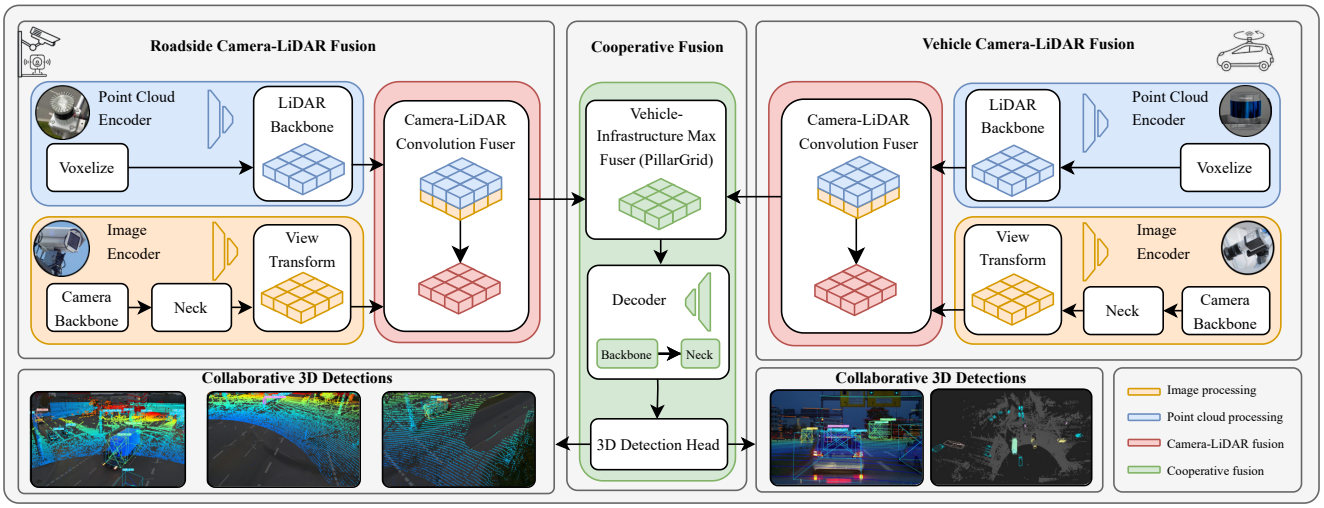


Figure 2. Our CoopDet3D framework is a multi-modal cooperative fusion system, comprising three distinct fusion pipelines. 1) The roadside camera-LiDAR fusion pipeline fuses three camera images and one LiDAR point cloud by extracting features and transforming them into a BEV representation. 2) The vehicle camera-LiDAR fusion pipeline fuses the vehicle camera feature map with the vehicle point cloud feature map using a convolutional fuser. 3) The vehicle and infrastructure feature maps are then fused by applying an element-wise max-pooling operation (Max Fuser). In the end, we use the TransFusion [2] 3D detection head to obtain 3D bounding box predictions.

and LiDAR) datasets are presented in [7, 15, 64], which are obtained from Infrastructure Perception Systems (IPS). Similarly, in [55], the authors provide a dataset consisting of only images taken from different viewpoints and under varying traffic conditions. These datasets provide a top-down view of a crowded intersection under different conditions and, as such, can overcome issues such as occlusions created by other vehicles and thereby have a higher number of object labels than onboard sensor-based datasets.

2.2. V2X datasets

V2X datasets exploit the information from multiple viewpoints to gain additional knowledge regarding the environments. In this way, they overcome the limitations of single viewpoint datasets such as occlusion, limited field of view (FOV), and low point cloud density.

DAIR-V2X dataset family [57] is one of the foremost cooperative multi-modal datasets introduced. It contains three subsets: an intersection, a vehicle, and a cooperative dataset. The cooperative dataset contains 464k 3D box labels belonging to 10 classes, making it one of the largest cooperative datasets. The V2X-Seq dataset [59] extends selected sequences of the DAIR-V2X dataset with track IDs and is partitioned into a sequential perception dataset (SPD) and a trajectory forecasting dataset. Despite these, the lack of specific information, such as the labeling methodology used, the exact models of the sensors deployed, the distribution of the classes, and the scenarios within the dataset, leads to uncertainty in the extendability and application of this dataset in varying conditions.

In V2V4Real [52], the authors propose a multi-modal cooperative dataset focusing only on V2V perception. Two vehicles equipped with cameras, LiDAR, and GPS/IMU integration systems are used to collect multi-modal sensor data for diverse scenarios. As opposed to all other V2X datasets, this focuses on V2V perception, and though it is of similar size to other cooperative datasets, it contains fewer classes and 3D bounding box information.

Simulated multi-agent perception datasets have been proposed in [30, 49, 51]. These datasets contain multi-modal sensor data (camera and LiDAR) obtained from roadside units (RSUs) and multiple ego vehicles, which enable collaborative perception. They use a combination of simulators such as SUMO [35], CARLA [16], and OpenCDA [51] for flow simulation, data retrieval, and V2X communication. However, the utility of the dataset is still limited due to the simulated nature of the data, and its extendability to real-life applications has not been studied in detail.

2.3. V2X perception models for object detection

The datasets presented above have been used to develop various models for a wide variety of tasks, with the majority focusing on 3D object detection. Different approaches have been taken depending on the availability and challenges, and these methods are grouped based on the number of nodes employed and the modalities used for detection.

Most 3D object detection models use multi-modal sensor data obtained from a single point of view, which is often an ego-vehicle. Due to the popularity and abundant availability of vehicle datasets [9, 19, 42], most models use images,

point cloud data, or both modalities. Image-based models were the pioneers in 3D object detection due to their low cost and simplicity, and both vehicular camera-based models [26, 48] and infrastructure camera-based models [54] have been proposed. LiDAR-based 3D object detection models [27, 65] became popular since LiDAR point clouds provide 3D depth information and are robust, especially in adverse weather conditions and nighttime scenarios. Fusion models combine the information obtained from both images and point clouds and have been shown to outperform the prior methods [63]. Single viewpoint fusion models use either vehicular camera and LiDAR [34, 44, 53] or infrastructure camera and LiDAR [63] for 3D object detection.

Cooperative perception models, which use data from multiple viewpoints, have been shown to overcome issues related to occlusion, which were often present in vehicular sensor-based models. V2I cooperative perception models [3, 4, 21, 37, 46, 49, 58] use the sensor data from both vehicles and infrastructure and V2V models [22, 41] communicate the sensor data between multiple vehicles. In this work, our cooperative multi-modal dataset is one contribution among others. Thus, while most of the prior works focus on unimodal cooperative perception using either LiDAR point clouds [3, 11] or camera images [22], we benchmark our dataset with CoopDet3D, a deep fusion based cooperative multi-modal 3D object detection model based on BEVFusion [22] and PillarGrid [3].

3 . TUMTraf-V2X Dataset

Our TUMTraf V2X Cooperative Perception Dataset focuses on challenging traffic scenarios and various day and nighttime scenes. The data is further annotated, emphasizing high-quality labels through careful labeling and high-quality review processes. It also contains dense traffic and fast-moving vehicles, which reveals the specific challenges in cooperative perception, such as pose estimation errors, latency, and synchronization. Furthermore, we provide sensor data from nine different sensors covering the same traffic scenes under diverse weather conditions and lighting variations. The infrastructure sensors are oriented in all four directions of the intersection to get a 360° view, which leads to better perception results. Finally, it contains rare events like traffic violations where pedestrians cross the road at a busy four-way intersection while the crossing light is lit red.

3.1. Sensor setup

Our TUMTraf V2X Cooperative Perception Dataset was recorded on an ITS system with nine sensors.
The infrastructure sensor setup is the following:

- 1x Ouster LiDAR OS1-64 (gen. 2), 64 vert. layers, 360° FOV, below horizon config., 10 cm acc. @120 m range
- 4x Basler ace acA1920-50gc, 1920×1200, Sony IMX174 with 8 mm lenses

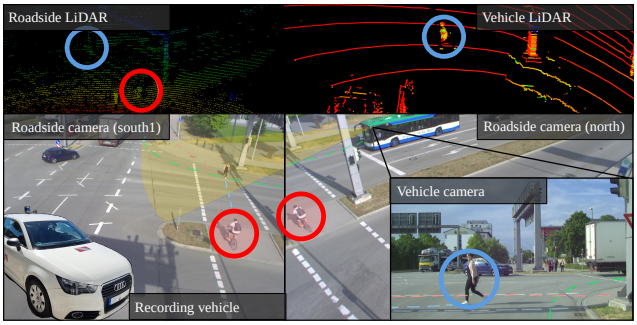


Figure 3. Demonstration of a possible V2X occlusion scenario. A pedestrian (blue) is crossing the road in front of the ego vehicle. An occluded bicycle is marked in red. The recording vehicle with the sensor setup is shown in the bottom left corner.

On the vehicle, the following sensors were used:

- 1x Robosense RS-LiDAR-32, 32 vert. layers, 360° FOV, 3 cm accuracy @200 m range
- 1x Basler ace acA1920-50gc, 1920×1200, Sony IMX174 with 16 mm lens
- 1x Emlid Reach RS2+ multi-band RTK GNSS receiver
- 1x XSENS MTi-30-2A8G4 IMU

3.2. Sensor calibration and registration

We synchronize the cameras and LiDARs in the spatial and temporal domain. First, we determine the intrinsic camera parameters and the radial and tangential image distortions by using a checkerboard target. We then calibrate the roadside LiDAR with the roadside cameras by picking 100-point pairs in the point cloud and camera image. Extrinsic parameters (rotation and translation) are calculated by minimizing the reprojection error of 2D-3D point correspondences [36]. We follow the same procedure for onboard camera-LiDAR calibration. Finally, we calibrate the onboard LiDAR to the roadside LiDAR. This spatial registration is done by first estimating a coarse transformation. We pick ten 3D point pairs in each point cloud and minimize their distance using the least squares method. Then, we apply the point-to-point Iterative Closest Point (ICP) algorithm [5] to get the fine transformation between the point clouds.

We label the vehicle and infrastructure point clouds after registering them. The coarse registration was done by measuring the GPS position of the onboard LiDAR and the roadside LiDAR. Then, we transform every 10th onboard point cloud to the coordinate system of the infrastructure point cloud. The fine registration was done by applying the point-to-point ICP to get an accurate V2I transformation matrix. All rotations of the point cloud frames in between are interpolated based on the spherical linear interpolation (SLERP) [40] method:

$$SLERP(q_0, q_1, t) = q_0(q_0^{-1}q_1)^t, \quad (1)$$

where q_0 and q_1 are the quaternions representing the rotations of the start and end frames and $t \in [0, 1]$. Translation vectors $\mathbf{T}_0$ and $\mathbf{T}_1$ were obtained using linear interpolation:

$$\mathbf{T}(t) = \mathbf{T}_0 + t(\mathbf{T}_1 - \mathbf{T}_0). \quad (2)$$

This dual interpolation strategy ensures that the estimated transformations between the frames are smooth and geometrically accurate, thus adhering closely to the actual movements of the vehicle over time.

3.3. Data selection and labeling

We selected the data based on challenging traffic scenarios, like U-turns, tailgate events, and traffic violation maneuvers. Besides the high traffic density of 31 objects per frame, we selected frames with high-class coverage. We selected 700 frames during sunny daytime and 100 frames during cloudy nighttime for labeling. The camera and LiDAR data were recorded into rosbag files at 15 Hz and 10 Hz, respectively. We extracted and synchronized the data based on ROS [38] timestamps and labeled it with our *3D BAT* (v24.3.2) annotation tool[1]. We improved the 3D BAT [62] baseline labeling tool to label 3D objects faster and more precisely with a one-click annotation feature. The annotators were instructed to label traffic participants while examining the images. Objects are still labeled, even if they have no 3D points inside, but are visible in the images. Extremities (e.g., pedestrian limbs) are included in the bounding box, but side mirrors of vehicles aren't. If a pedestrian carries an object, that object is included in the bounding box. If two or more pedestrians are carrying an object, only the box of one will include the object. After labeling, each annotator checked the work of other annotators manually frame-by-frame. When errors were found, the original annotator was notified, and they fixed it. This helps ensure that the labels in our dataset are high quality.

3.4. Data structure and format

We record eight different scenes, each 10 sec. long, from vehicle and infrastructure perspectives using nine sensors and split the data into a train (80%), val. (10%), and test (10%) set. We use stratified sampling to distribute all sets' object classes equally (see Fig. 6a). Labels are provided in the ASAM OpenLABEL [20] standard.

3.5. Dataset development kit

We provide a dev-kit to work with our dataset. In addition to generating the data statistics, it provides modules for multi-class stratified splitting (train/val/test), point cloud registration, loading annotations in OpenLABEL format, evaluation of detection and tracking results, pre-processing steps such as point cloud filtering, and post-processing such as bounding box filtering. The statistics from Figures 4, 5, and 6 were created using our dataset dev kit. It also contains modules to convert the labels from OpenLABEL to KITTI or our custom nuScenes format with timestamps instead of tokens and vice versa. This dev kit enables users of popular datasets to migrate their models and make them compatible with our dataset format. We release our dev kit[2] under the MIT license and the dataset under the Creative Commons (CC) BY-NC-ND 4.0 license.

Table 2. Evaluation results (mAP_{BEV} and mAP_{3D}) of CoopDet3D on our TUMTraf-V2X test set in south2 FOV.

Config.		$\mathbf{mAP_{BEV}}\uparrow$	$\mathbf{mAP_{3D}}\uparrow$			
Domain	**Modality**		**Easy**↑	**Mod.**↑	**Hard**↑	**Avg.**↑
Vehicle	Camera	46.83	31.47	37.82	30.77	30.36
Vehicle	LiDAR	85.33	85.22	76.86	69.04	80.11
Vehicle	Cam+LiDAR	84.90	77.60	72.08	73.12	76.40
Infra.	Camera	61.98	31.19	46.73	40.42	35.04
Infra.	LiDAR	92.86	86.17	88.07	75.73	84.88
Infra.	Cam+LiDAR	92.92	87.99	**89.09**	**81.69**	87.01
Coop.	Camera	68.94	45.41	42.76	57.83	45.74
Coop.	LiDAR	93.93	92.63	78.06	73.95	85.86
Coop.	Cam+LiDAR	**94.22**	**93.42**	88.17	79.94	**90.76**

Table 3. Evaluation results of infrastructure-only CoopDet3D vs. InfraDet3D [63] on TUMTraf Intersection test set [64].

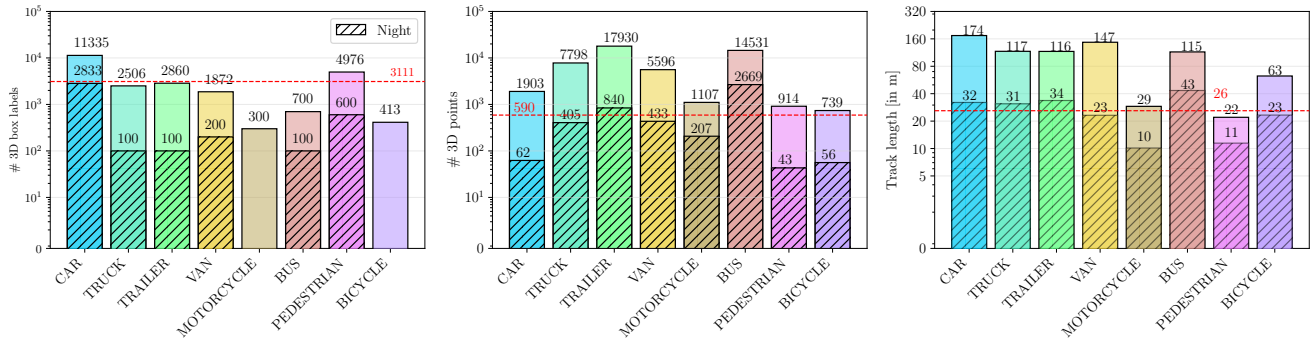
			$\mathbf{mAP_{3D}}\uparrow$			
Config.	**FOV**	**Mod.**	**Easy**↑	**Mod.**↑	**Hard**↑	**Avg.**↑
InfraDet3D	south 1	LiDAR	75.81	47.66	**42.16**	55.21
CoopDet3D	south 1	LiDAR	**76.24**	**48.23**	35.19	**69.47**
InfraDet3D	south 2	LiDAR	38.92	46.60	**43.86**	43.13
CoopDet3D	south 2	LiDAR	**74.97**	**55.55**	39.96	**69.94**
InfraDet3D	south 1	Cam+LiDAR	67.08	31.38	35.17	44.55
CoopDet3D	south 1	Cam+LiDAR	**75.68**	**45.63**	**45.63**	**66.75**
InfraDet3D	south 2	Cam+LiDAR	58.38	19.73	33.08	37.06
CoopDet3D	south 2	Cam+LiDAR	**74.73**	**53.46**	**41.96**	**66.89**

Table 4. Ablation study on cooperative 3D object detection with 11 combinations of camera and LiDAR backbones. The best trade-off between speed and accuracy is highlighted in gray.

Backbone Configuration	$\mathbf{mAP_{BEV}}\uparrow$	**FPS**↑	**VRAM**↓
VoxelNet non-deterministic + SwinT	93.47	6.30	6.69 GiB
VoxelNet non-deterministic + YOLOv8 s	92.94	7.24	6.39 GiB
VoxelNet Torchsparse + SwinT	93.51	8.84	4.61 GiB
VoxelNet Torchsparse + YOLOv8 s	92.94	10.66	**4.28 GiB**
VoxelNet Torchsparse + YOLOv8 s (retrained)	94.31	10.66	**4.28 GiB**
PointPillars 512 + Swin T	**94.43**	9.00	4.94 GiB
PointPillars 512 + YOLOv8 s	94.27	11.14	4.63 GiB
PointPillars 512 + YOLOv8 s (retrained)	94.25	11.14	4.63 GiB
PointPillars 512_2x + Swin T	92.79	9.06	4.94 GiB
PointPillars 512_2x + YOLOv8 s	94.16	**11.20**	4.63 GiB
PointPillars 512_2x + YOLOv8 s (retrained)	94.22	**11.20**	4.63 GiB

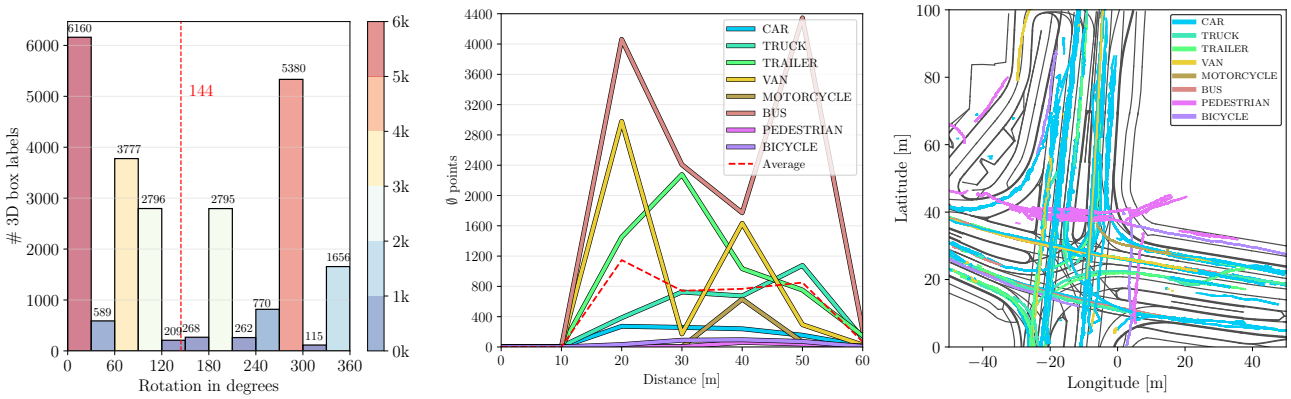
[1] https://github.com/walzimmer/3d-bat

[2] https://github.com/tum-traffic-dataset/tum-traffic-dataset-dev-kit



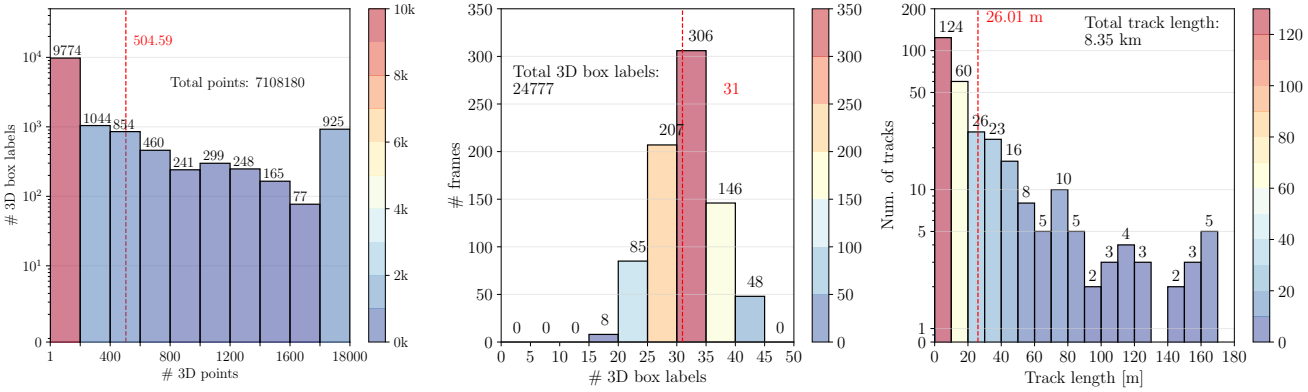
(a) Distribution of objects between day and night. (b) Avg. and max. num. of 3D points for each class. (c) Avg. and max. track length for all classes.

Figure 4. Our TUMTraf-V2X dataset (version 1.0) contains 25k 3D box labels in total and is balanced among eight different object classes. (a) Cars (11,203) and pedestrians (4,781) are highly represented in the dataset. (b) 3D box labels contain on average 590 points inside, which shows the density of the labeled objects. The BUS class has the highest point density. (c) All traffic participants are tracked for 26 m on average. Buses have the highest average track length of 43 m, whereas the CAR class contains the max. track length of 173.95 m.



(a) Visualization of rotations (yaw). (b) 3D points grouped by distance. (c) BEV visualization of tracks.

Figure 5. Our dataset was recorded at a crowded intersection with many left and right turns. (a) Most of the vehicles (6,160) are driving in the east direction (0 degree). (b) 3D boxes were labeled up to 200 m range and are very dense between 10 and 60 m. (c) The visualization of BEV tracks shows where pedestrians and bicycles are crossing the road.



(a) Histogram of 3D points. (b) Histogram of 3D box labels. (c) Histogram of track lengths.

Figure 6. (a) Most 3D box labels contain between 1 and 200 3D points inside, with an average of 505 3D points, excluding empty boxes. Objects that were close to both LiDAR sensors even contained up to 18k 3D points. (b) Frames contain between 20 and 45 traffic participants, with an average of 31. (c) Objects were tracked up to 180 m and the average track length is 26 m.

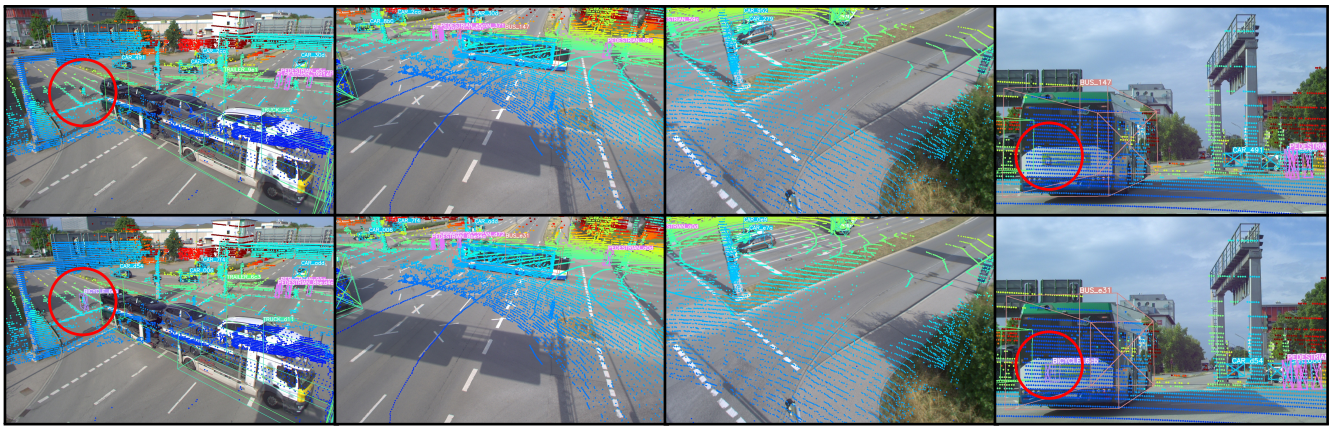


Figure 7. Qualitative results on the TUMTraf-V2X Cooperative Perception test set. The first row shows the inference results of the onboard (vehicle-only) camera-LiDAR fusion with 23 detected objects. In the second row, the results of the cooperative vehicle-infrastructure camera-LiDAR fusion are visualized. Here, the 25 traffic participants could be detected with the support of roadside sensors.

4 . Benchmark

We propose CoopDet3D, an extension of BEVFusion [34] and PillarGrid [3] for deep cooperative multi-modal 3D object detection and benchmark it on our dataset.

4.1. Evaluation metrics

The accuracy is measured in terms of the mean average precision (mAP). Two types of mAP measures are used: BEV mAP considers the BEV center distance, and the results are obtained using the same evaluation methodology used in BEVFusion, which in turn uses the evaluation protocol of nuScenes [9]. Similarly, the 3D mAP measure considers the intersection in 3D, and the results are obtained using the evaluation script of our TUM Traffic dataset Devkit. The runtime is evaluated using frames per second (FPS) as the metric and the results were obtained by measuring the time needed by the model to run one full inference, including data preprocessing and voxelization. The first five iterations are skipped as a warmup since they are usually considerably slower than the average. Finally, the complexity of the model is measured in terms of the maximal VRAM usage across all GPUs during training and testing.

4.2. CoopDet3D model

Our *CoopDet3D* uses a BEVFusion-based backbone for camera-LiDAR fusion on the vehicle and the infrastructure sides separately to obtain the vehicle and infrastructure features. The best backbone for image and point cloud feature extraction was chosen through multiple ablation studies. Then, inspired by the method proposed by PillarGrid [3], an element-wise max-pooling operation is proposed to fuse the resulting fused camera-LiDAR features of vehicle and infrastructure together. Finally, the detection head from BEVFusion is used for 3D detection from the fused feature. The architecture of *CoopDet3D* is shown in Fig. 2.

First, we disable the camera feature extraction nodes and train the LiDAR-only model for 20 epochs. Then, we use pre-trained weights for the cooperative model and fine-tune the entire model for eight further epochs. Hyperparameter tuning revealed that the default hyperparameters of BEVFusion [34] gave the best results, and such were not modified. The preprocessing steps are also the same as the BEVFusion, but we change the point cloud range to $[-75, 75]$ in the x- and y-scale and $[-8, 0]$ in the z-scale since the dataset used in this case is different. Furthermore, we use 3x NVIDIA RTX 3090 GPUs with 24 GB VRAM for training and a single GPU for evaluation. We open-source our model and provide pre-trained weights[3].

4.3. Experiments and ablation studies

The objective of these experiments is to highlight the importance of our V2X multi-viewpoint dataset as opposed to single-viewpoint datasets. As such, we conduct multiple experiments and ablation studies with data obtained from each viewpoint and compare the results on the proposed model.

Cooperative perception compared to single-viewpoint

We conduct multiple experiments with all possible combinations of a) viewpoints: vehicle-only, infrastructure-only, cooperative, and b) modalities: camera-only, LiDAR-only, and camera-LiDAR fusion. Table 2 shows the mAP achieved by CoopDet3D for each of these combinations.

We observe that the results follow a general pattern of cooperative performance being better than infrastructure-only, which is, in turn, better than vehicle-only. Furthermore, fusion models perform better than LiDAR-only models, which in turn are better than camera-only models. Figure 7 shows qualitative results between our vehicle-only

[3] https://github.com/tum-traffic-dataset/coopdet3d

camera-LiDAR fusion model and our cooperative vehicle-infrastructure camera-LiDAR fusion model. Again, we observe from these samples that the cooperative perception model is able to detect 25 traffic participants, whereas the vehicle-only model is only able to detect 23 objects due to occlusions and a limited field of view.

Deep fusion compared to late fusion

Next, we compare our proposed CoopDet3D model to the current SOTA camera-LiDAR fusion method on the TUMTraf Intersection test set [64], InfraDet3D [63]. The proposed method uses deep fusion, whereas the InfraDet3D method is a late fusion method. Table 3 shows the performance of our model against InfraDet3D, and the results show that the proposed deep fusion method outperforms the SOTA late fusion model in all metrics, except in the hard difficulty in LiDAR-only mode. Furthermore, Figure 8 shows two sample images taken during day and nighttime, wherein deep fusion again outperforms late fusion. South 1 and South 2 refer to sensors covering different FOVs.

We note that these experiments were conducted in an offline setting, disregarding other considerations for simplicity. However, when deploying it in real life, factors such as the transmission bandwidth should also be considered. Since we observed that deep feature fusion generally leads to higher efficacy, the V2I transmissions should contain these features instead of infrastructure bounding boxes.

Model performance with different backbones

As an ablation study, we present the results of the experiments to find the best backbone and model configuration for the cooperative camera-LiDAR fusion model. For the camera backbone, SwinT [33] and MMYOLO's [13] implementation of YOLOv8 [24] were considered. For the LiDAR backbone, VoxelNet [61] and PointPillars [27] were considered. In addition, VoxelNet was implemented with two different backends, namely SPConv v2 and Torchsparse [43]. For PointPillars, two grid sizes are considered 512×512 for both train and test grids (PointPillars 512) and 512×512 train grid with 1024×1024 test grid (PointPillars 512_2x). The results of these experiments are shown in Table 4.

The results show that only models that use any combination of VoxelNet Torchsparse, both PointPillars variants, and YOLOv8 are able to run above 10 FPS. From these configurations, we choose PointPillars 512_2x with YOLOv8 as the best configuration for all the above experiments as it achieves the best results across all the ablation studies. This is a promising result since we also know that this backbone configuration is able to run in real-time (11.2 FPS) on an RTX 3090 without using TensorRT acceleration.

An interesting observation is that utilizing pre-trained weights for transfer learning of YOLOv8 is not always beneficial, as the results from PointPillars 512 + YOLOv8 s show. This is likely because the pre-trained weights were from MS COCO [31], and they have a very different data domain compared to our dataset. Since MS COCO is also much larger than our dataset in terms of camera images, retraining harms the performance of the model slightly.

Figure 8. Qualitative results of our CoopDet3D (left) and the InfraDet3D (right) model on the TUMTraf Intersection test set during day and nighttime. Detected objects marked with a red circle were classified correctly by CoopDet3D.

In terms of efficiency, the goal of these experiments was to verify that the proposed CoopDet3D model with the best configuration provides the highest accuracy while also being able to run in real-time (minimum of 10 Hz). Furthermore, it should also be feasible to train the model on a high-performance GPU and perform inference on a mid-range consumer GPU deployable on an edge device. The results concerning the VRAM usage during inference show that the complexity of the model makes this feasible.

5 . Conclusion and future work

This work proposes the TUMTraf-V2X dataset, a multimodal multi-view V2X dataset for cooperative 3D object detection and tracking. Our dataset focuses on challenging traffic scenarios at an intersection and provides views from the infrastructure and the ego vehicle. To benchmark the dataset, we propose CoopDet3D – a baseline model for cooperative perception. Experiments show that cooperative fusion leads to higher efficacy than its unimodal and single-view camera-LiDAR fusion counterparts. Furthermore, cooperative fusion leads to an improvement of +14.3 3D mAP compared to vehicle-only perception, highlighting the need for V2X datasets. Finally, we provide our *3D BAT* v24.3.2 labeling tool and dev kit to load, parse, and visualize the dataset. It also includes modules for pre- and postprocessing and evaluation. Future efforts will integrate this platform into online environments, enabling a broader range of infrastructure-based, real-time perception applications.

Acknowledgment

This research was supported by the Federal Ministry of Education and Research in Germany within the *AUTOtech.agil* project, Grant Number: 01IS22088U.

TUMTraf V2X Cooperative Perception Dataset

Supplementary Material

https://tum-traffic-dataset.github.io/tumtraf-v2x

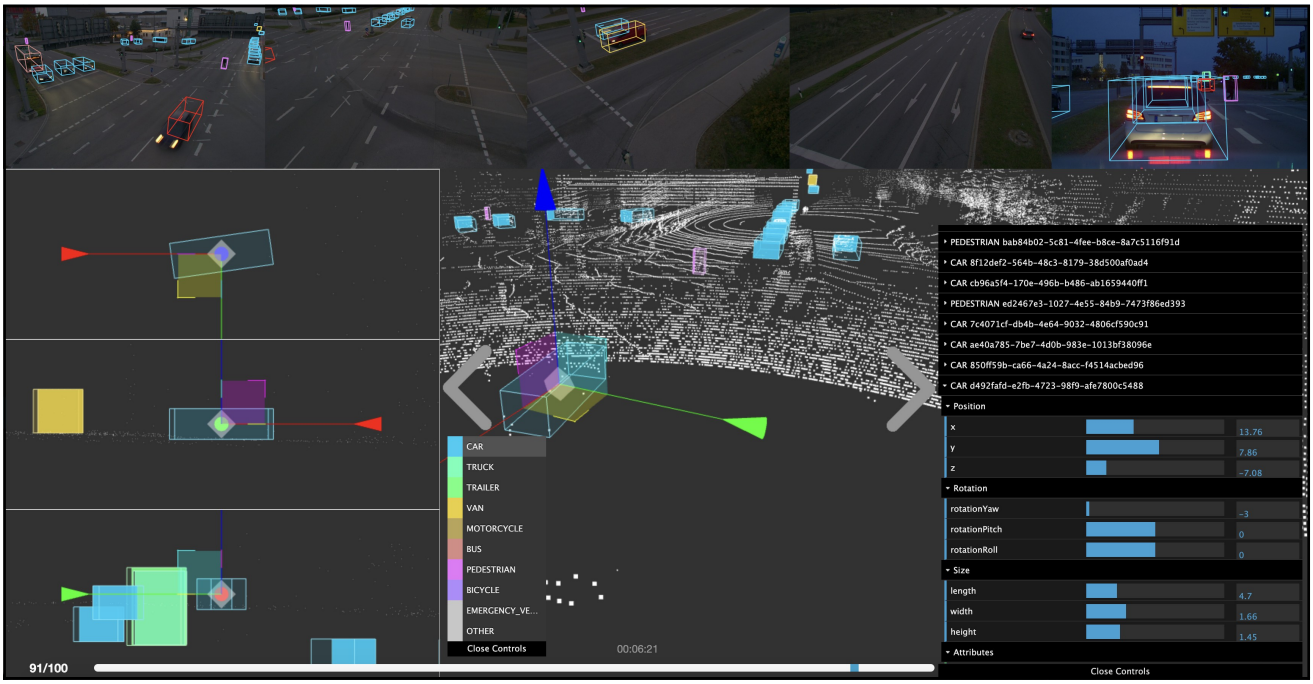


Figure 9. Visualization of our web-based *3D BAT* (v24.3.2) labeling tool. It shows the registered point cloud and five camera images on the top. On the left side, there are three helper views: top-down view, side view, and front view. The control pane on the right side contains a download button, an undo button, a drop-down menu to switch between a perspective (3D) and orthographic (BEV) view, a slider to change the point size, a drop-down menu to choose the dataset and sequence, some checkboxes for filtering the scene and hiding other annotations, a button to copy labels to the next frame, an auto-label button, a button for active learning, an interpolation button, and a reset button. In the bottom right corner, all labeled objects are displayed. Each object can be translated, scaled, and rotated using sliders or keyboard shortcuts. The scaling of an object will change the dimensions in all frames.

Contents

A . Task Definition

A.1. Detection and tracking

Detection and tracking are two crucial perception tasks for autonomous driving. In 3D object detection, the surrounding objects are located with their 3D position, dimensions (length, width, height), and rotation at each timestamp. In multi-object tracking (MOT), the correspondences between different objects are found across timestamps. Objects are associated temporally and given a unique track ID. The final detection and tracking output is a series of associated 3D boxes in each frame.

A.2. Cooperative fusion

The cooperative fusion approach combines data from several sensors from different perspectives to optimize the detection and tracking performance. Data from roadside cameras and LiDARs is fused with onboard camera and LiDAR sensor data to prevent occlusions.

B . Problem statement

We consider a cooperative perception system with roadside and vehicle sensors symbolized by r_s $s \in [C, L]$ and v_s $s \in [C, L]$ notations, respectively. The cooperative system introduced in this work uses three infrastructure cameras r_{Ci} $i \in 1, 2, 3$ where i denotes the camera IDs, an infrastructure LiDAR r_L, one onboard vehicle camera v_C and one onboard LiDAR v_L. Consequently, the vehicle sensors produce a set of images $v_I(\hat{t})$ and point clouds $v_P(\hat{t})$, and the infrastructure sensors produce a set of images $r_{Ii}(t')$, and point clouds $r_P(t')$. Here, $\hat{t}$ and t' denote the vehicle and infrastructure data timestamps respectively. Note that a small synchronization error is still present though the infrastructure and roadside sensors are all synchronized to the same NTP time server. The average difference in timestamps between these two systems $\mathbf{E}[\hat{t} - t']$ is 24.91 ms and the two data sources are matched in our proposed dataset using the nearest neighbor matching algorithm.

The objective of cooperative 3D detection is to predict 3D bounding boxes of objects given a set of multi-modal multi-viewpoint data. Our proposed cooperative detection model takes the set of images and point clouds as the input $X(t) = [v_I(t), v_P(t), r_{I1}(t), r_{I2}(t), r_{I3}(t), r_{P1}(t)]$ at a given time t and predicts the 3D bounding boxes as the output $\tilde{Y}(t)$. Here, t denotes the shared timestamp after the matching algorithm. In addition to identifying the boxes' position, dimensions, and orientation, the proposed model also predicts the class of the corresponding object. Thus, we can represent the task of 3D object detection as:

$$\min \mathop{\mathbb{E}}_{y_j \in Y(t)} \left[\min_{\tilde{y}_k \in \tilde{Y}(t)} d_\theta \left(y_j, \tilde{y}_k \right) \right] \tag{3}$$

where $Y(t) = [y_1(t), y_2(t), ...]$ is the set of ground truth 3D box labels at time t, and $\tilde{Y}(t) = [\tilde{y}_1(t), \tilde{y}_2(t), ...]$ are the corresponding predicted 3D boxes. $d_\theta(y_j, y_k)$ is a parameterized discriminator function which measures the error between ground truth 3D label y_j and the predicted 3D box y_k. Thus, our objective is to reduce the total error.

C . Data anonymization

We anonymize all our camera raw images $I = [v_I, r_{I1}, r_{I2}, r_{I3}, r_{I4}]$ in the roadside and vehicle domain by obfuscating all license plate numbers and faces. We use a medium *YOLOv5* model [23] for this purpose, which was pre-trained on 1080p images with labeled license plates and faces. During training, mosaic augmentation was applied to teach the model to recognize objects in different locations without relying too much on one specific context. At inference, we downscale the input images I from a 1920×1200 resolution to 640×400 and pad the extra space to 640×640. A score threshold of 0.1 worked best to detect all private information. We set the granularity of the blurring filter to a blur size of 6 for the detected regions and set the ROI multiplier to 1.1.

D . Further related work

This section compares our proposed *3D BAT* v24.3.2 annotation tool and development kit to similar open-source tools.

D.1. Annotation tools

This work proposes our annotation tool *3D BAT* v24.3.2, which supports combining LiDAR point clouds and simultaneously labels both the point clouds and images from multiple views.

3D BAT [62] is an open-source, web-based annotation framework designed for efficient and accurate 3D annotation of objects in LiDAR point clouds and camera images. With this tool, 2D and 3D box labels can be obtained, as well as track IDs. Its key features include semi-automatic labeling using interpolation of objects between frames. Labeled 3D boxes are automatically projected into all camera images, which requires extrinsic camera-LiDAR calibration data. Selected objects are displayed in a bird's eye view, side view, and front view, in addition to a perspective and orthographic view.

SUSTechPoints [28] is a multi-modal 3D object annotation tool. It first allows the addition of 3D bounding boxes in point clouds and then updates them in six degrees of freedom. It furthermore allows updating bounding boxes' type, attributes, and ID to create labeled datasets for detection and tracking tasks. It also allows users to visualize these boxes projected onto multiple camera images and lets the user enable or disable the point clouds and images for clear visualization. One major advantage of *SUSTechPoints* is that it

Table 5. Comparison of 3D annotation tools. ● Feature provided ○ Feature unknown ● Feature not provided

Tool	3D BAT [62]	LATTE [45]	SAnE [1]	SUSTech POINTS[28]	Label Cloud[39]	ReBound [10]	PointCloud Lab[17]	Xtreme1 [18]	**3D BAT (Ours)***
Year	2019	2020	2020	2020	2021	2023	2023	2023	2023
Support V2X	-	-	-	-	-	-	-	✓	✓
2D/3D cam.+LiDAR fusion	✓	✓	-	✓	✓	✓	-	✓	✓
AI assisted labeling	✓	✓	✓	✓	-	✓	✓	✓	✓
Batch-mode editing	-	-	-	✓	-	-	-	✓	✓
Interpolation mode	-	-	-	✓	-	-	-	-	✓
Active learning support	-	-	-	✓	-	-	-	-	✓
Label custom attributes	-	-	-	✓	-	✓	(?)	✓	✓
3D tracking	✓	✓	✓	✓	-	-	✓	-	✓
Support multiple cameras	✓	-	-	✓	-	✓	-	✓	✓
HD Maps	-	-	-	✓	-	-	-	✓	✓
Web-based	✓	-	-	✓	-	-	-	✓	✓
3D navigation	✓	✓	✓	✓	-	✓	✓	✓	✓
3D transform controls	✓	✓	✓	✓	-	✓	✓	✓	✓
Side views (top/front/side)	✓	-	✓	✓	-	✓	-	✓	✓
Perspective view editing	✓	✓	✓	✓	✓	✓	✓	✓	✓
Orthographic view editing	✓	✓	✓	-	-	✓	✓	✓	✓
Object coloring	✓	-	✓	✓	-	✓	✓	✓	✓
Focus mode	-	-	-	✓	-	-	✓	✓	✓
Support JPG/PNG files	-	(?)	(?)	✓	-	-	-	(?)	✓
Keyboard-only support	-	-	-	-	-	-	-	-	✓
Offline annotation support	-	-	-	-	-	-	-	-	✓
OpenLABEL support	-	-	-	-	-	-	-	-	✓
Open-source	✓	✓	✓	✓	✓	✓	-	✓	✓
Github stars	529	374	62	670	461	20	-	542	543
Citations	46	36	16	33	16	0	2	0	54
License	MIT	Apach. 2.0	Apach. 2.0	GPL 3.0	GPL 3.0	Apach. 2.0	-	Apach. 2.0	MIT

*We use the latest release of 3D BAT version v24.3.2.

enables auto box fitting based on the point cloud shape, but the accuracy of the fitted box is highly dependent on the point cloud density.

labelCloud [39] is a domain-agnostic, lightweight tool designed specifically to label 3D objects. It offers two labeling modes namely picking and spanning. In the picking mode, objects with known sizes can be quickly adjusted. The spanning mode simplifies labeling by reducing the process to four clicks. Box dimensions and orientations of objects on flat surfaces can be efficiently defined.

ReBound [10] is an open source 3D bounding box annotation tool designed to utilize active learning. It supports loading, visualizing, and extending existing datasets like nuScenes [9], Waymo [42] or Argoverse 2.0 [47]. Model predictions can be analyzed and corrected in a 3D view and exported to specific formats.

PointCloudLab [17] leverages virtual and augmented reality (VR/AR) devices for 3D point cloud annotation. The annotator utilizes the controller of a HTC Vive to perform object-level annotations in the 3D point cloud. The immersive visual aid accelerates the labeling speed, improves the labeling quality, and enhances the labeling experience.

The *Xtreme1* [18] labeling tool provides most of the functionalities of *SUSTechPoints*. In addition to providing automated 3D labeling, it also provides support for automated 2D detection and segmentation tasks. Furthermore, it also supports multi-view point cloud data as the input. The tool also provides an interface for identifying specific errors in the labeling process, and a mechanism to evaluate different models on the labeled dataset. Moreover, it uses modern cloud-based standards, databases, Kubernetes for managing containers, and GitLab CI automation.

D.2. Development kits

OpenCOOD [50] is an open cooperative detection framework for autonomous driving which supports popular simulated datasets such as OPV2V [51] and V2XSet [49]. Like the development kit proposed in this work, *OpenCOOD* allows data preparation, pre/post-processing, and visualization. Furthermore, it also supports training and testing different benchmark models on these simulated datasets. However, the *OpenCOOD* development kit only currently supports simulated datasets. Its full functionality is also limited to LiDAR-only cooperative perception, and images are only used for visualization. V2V4Real [52] extends the OpenCOOD development kit, to support real-world data and ad-

Table 6. Tracking results of SORT and PolyMOT on drive_41. P = Precision, R = Recall, MT = Mostly Tracked, PT = Partially Tracked, ML = Mostly Lost, FM = Track Fragmentations

Tracker	IDP↑	IDR↑	IDF1↑	Recall↑	Precision↑	GT	MT↑	PT↑	ML↓	FP↓	FN↓	IDS↓	FM↓	MOTA↑	MOTP↓
SORT* [6]	36.313	21.029	26.634	43.235	74.657	3400	5	**18**	11	499	1920	439	110	15.647	**100.185**
PolyMOT [29]	**68.416**	**42.559**	**52.475**	**46.735**	**75.130**	3400	**8**	15	**11**	**526**	**1811**	**13**	**30**	**30.882**	102.288

* We modify the SORT tracker to track objects in 3D.

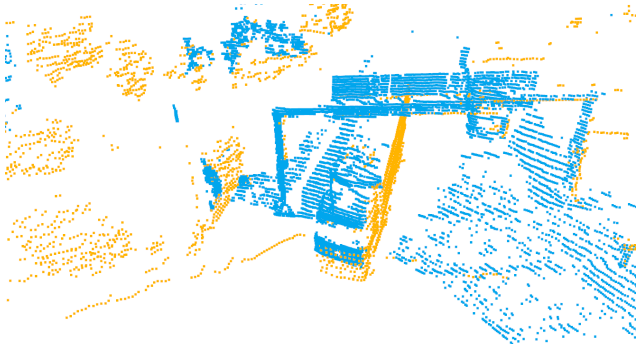


Figure 10. Point cloud registration results of an onboard LiDAR point cloud (orange) and a roadside LiDAR point cloud (blue).

ditional perception tasks. Furthermore, data augmentation is also an additional feature that can be enabled when training the model.

Furthermore, the DAIR V2X [57], proposes their own development kit, which provides data visualization and training tools. However, the access to the dataset is limited geographically. Other development kits, such as the Nuscenes devkit [8] and Rope3D devkit [55], only support unimodal or single-view point datasets.

In comparison, our proposed development kit allows all the aforementioned functionalities in both image and LiDAR modes. Furthermore, our development kit contains modules for multi-modal cooperative data augmentation, while the model training and testing depend on the *mmdetection* framework [12].

E . Point cloud registration details

We first measure the GPS position (latitude and longitude) of the onboard LiDAR and the roadside LiDAR and convert it to UTM coordinates. For the coarse registration, we transform every 10th onboard point cloud P_V to the infrastructure point cloud P_I coordinate system using the initial transformation matrix shown in Eq. 4.

$$T_{VI}^{0} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The transformation matrix T_0^{VI} contains as $3x3$ rotation matrix R obtained by the IMU sensor and a $3x1$ translation vector $\vec{t}$ obtained by the GPS device. We then apply the point-to-point ICP for the fine registration to get an accurate V2I transformation matrix T_{VI}.

$$P_{VI} = P_I \oplus (P_V \cdot T_{VI}) \tag{5}$$

Fig. 10 shows the point cloud registration results in two colors. The vehicle point cloud is displayed in orange, and the infrastructure point cloud is displayed in blue. We get an RMSE value of 0.02 m, which shows how well the point clouds were registered.

F . Dataset labeling

We provide a web-based labeling platform *3D BAT* v24.3.2 to facilitate the development of V2X perception. It provides a one-click annotation feature to fit an oriented bounding box to a 3D object. It contains an interpolation mode that reduces the labeling time significantly and lets the user visualize the HD Map, which is highly beneficial for positioning 3D box labels accurately within lanes. The user interface of *3D BAT* v24.3.2 is split into two main views: the upper portion displays the camera images captured by both infrastructure and vehicle-mounted cameras, while the lower portion renders the registered point cloud data obtained from the roadside and onboard LiDARs. The annotator first navigates the point cloud to identify objects of interest. Upon selecting an object, boxes are enclosed around it. These boxes are color-coded according to the object category (e.g., car, truck, trailer, van, motorcycle, bus, pedestrian, bicycle, and others) to allow for easy differentiation. After placing the 3D bounding box, they cross-check the predicted 2D bounding boxes in the camera images to ensure their correctness. Additional attributes can be modified and specified for each object on the right-hand side.

G . Implementation details

Here, we provide detailed information about the training schedule and the hyperparameters. We train our *CoopDet3D* model in two stages. In stage one, we pre-train the *PointPillars* backbone on onboard and roadside point clouds for 20 epochs. Then, in stage two, we finetune the model for eight further epochs on cooperative camera and LiDAR data. For the detection head, we use *TransFusion* [2] to obtain 3D bounding box predictions. To calculate the matching cost C_{match}, it uses a weighted binary cross

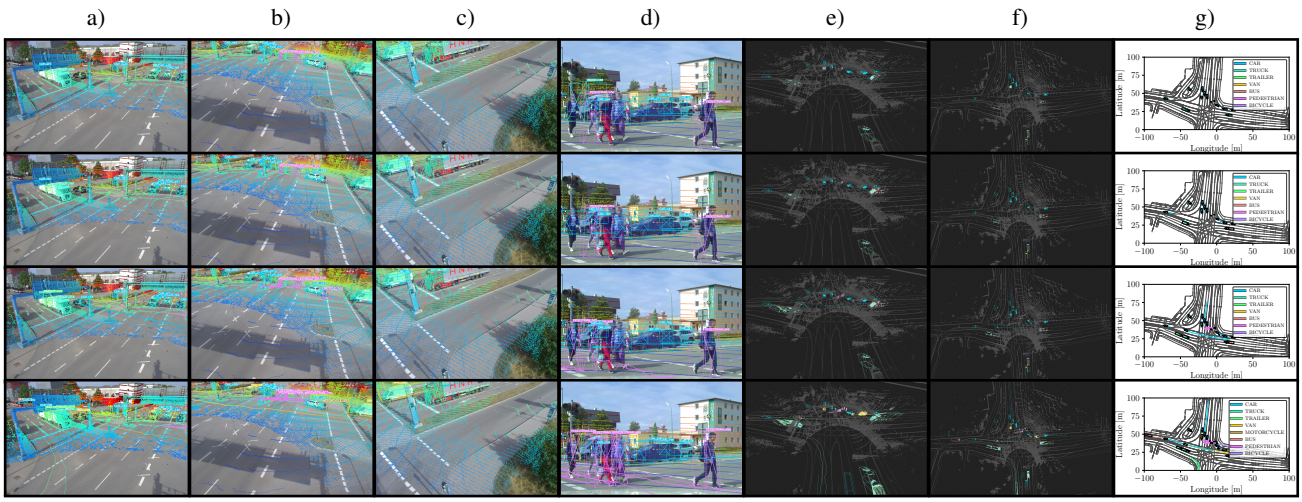


Figure 11. Tracking results on **drive_42** test sequence of the *TUMTraf-V2X* dataset. From top to bottom: *CoopDet3D* detections, *CoopDet3D* detections tracked by *SORT*, *CoopDet3D* detections tracked by *PolyMOT*, ground truth. a-c) Tracking results projected into roadside camera images. d) Tracking results visualized in vehicle camera. e) Visualization of tracks in a point cloud and the HD map. f) Bird's eye view projection of tracks in a point cloud and the HD map. g) Visualization of all detected classes and their tracks on an HD map.

Table 7. Evaluation results (mAP_{BEV} and mAP_{3D}) of *CoopDet3D* on our *TUMTraf-V2X* test set in south1 FOV.

Config.		$\mathbf{mAP_{BEV}}\uparrow$	$\mathbf{mAP_{3D}}\uparrow$			
Domain	**Modality**		**Easy**↑	**Moderate**↑	**Hard**↑	**Avg.**↑
Vehicle	Camera	46.83	39.31	12.42	4.29	35.02
Vehicle	LiDAR	85.33	77.30	31.26	53.76	76.68
Vehicle	Cam+LiDAR	84.90	77.29	34.29	39.71	76.19
Infra.	Camera	61.98	41.13	15.64	1.35	37.09
Infra.	LiDAR	92.86	82.16	45.14	46.56	81.07
Infra.	Cam+LiDAR	92.92	**85.43**	49.10	49.56	84.13
Coop.	Camera	68.94	52.04	29.26	10.28	49.81
Coop.	LiDAR	93.93	84.61	50.00	53.78	**84.15**
Coop.	Cam+LiDAR	**94.22**	84.50	**51.67**	**55.14**	84.05

entropy loss L_{cls}, a weighted L_1 loss for the 3D box regression $\mathcal{L}_{reg}$, and a weighted IoU loss $\mathcal{L}_{IoU}$ [60] (see Eq. 6).

$$C_{match} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{reg} + \lambda_3 \mathcal{L}_{IoU}, \tag{6}$$

where: λ_1, λ_2, λ_3 are the coefficients for the individual cost terms. Given all matched pairs, a focal loss [32] is computed for the final classification. A penalty-reduced focal loss [56] is used for the heatmap prediction.

We use the following hyperparameters for training: the *AdamW* optimizer with a learning rate of $1 \times e^{-4}$ and a weight decay of 0.01, a batch size of 4, a dropout rate of 0.1, the *ReLU* activation function, and cyclic momentum. We use the BEV encoder to transform the image into a BEV representation of 512×512 size. The point clouds are cropped to the following range: $[-75, 75]$ m for the X and Y axis, and $[-8, 0]$ m for the Z axis. For training, we use 3 x NVIDIA RTX 3090 GPUs.

H . Metrics

This section presents the evaluation metrics for two main tasks in V2X perception, i.e. the *Cooperative 3D Object Detection* (C3DOD) task and the *Cooperative Multiple Object Tracking* (CMOT) task. Notably, we adopt the mainstream metrics for the cooperative perception evaluation to make fair comparisons with the vehicle-only and infrastructure-only algorithms.

H.1. 3D Object Detection

As the most commonly utilized metric in 3D object detection tasks, *mean Average Precision* (mAP) (Eq. 7) takes the mean value of *Average Prevision* (AP) generally over the categories $\mathcal{C}$ of interest. We follow the approach of positive sample matching, introduced in *nuScenes* [9], leveraging 2D distance thresholds $\mathcal{D}$ on the ground plane between ground truth and prediction center positions, instead of using the intersection over union (IoU), to define a match (true positive). We match predictions with ground truth objects with the smallest center distance up to a certain threshold. For a given match threshold we calculate the *Average Precision* (AP) by integrating the recall-precision curve for recall and precision > 0.1. We finally average overmatch thresholds of $\mathcal{D} = \{0.5, 1, 2, 4\}$ meters and compute the mean across all classes.

$$mAP = \frac{1}{|\mathcal{C}||\mathcal{D}|} \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} AP_{c,d} \tag{7}$$

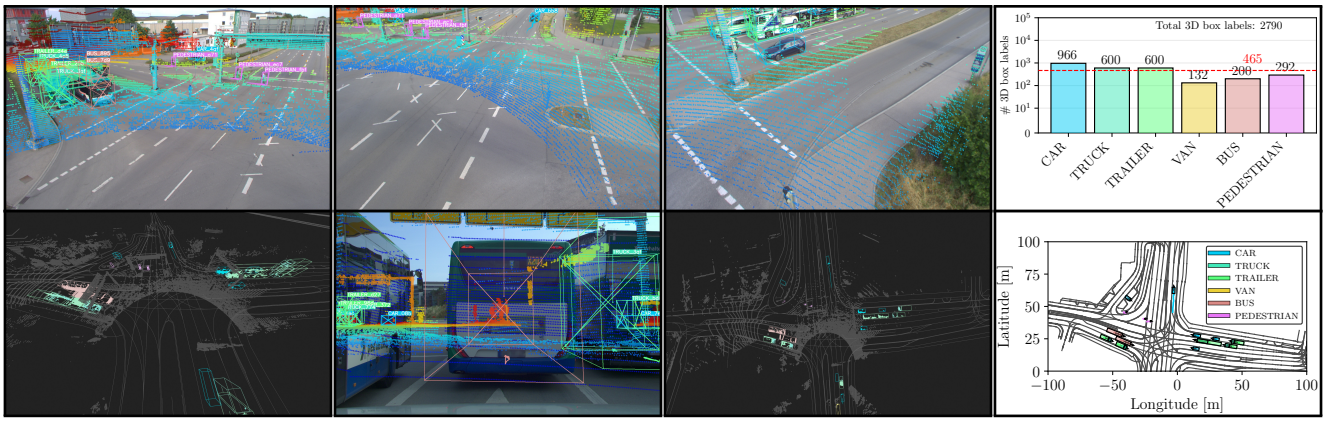


Figure 12. Visualization of **drive_07** of the *TUMTraf-V2X* dataset. In this example, the ego vehicle is occluded by two busses and two large trucks. The roadside sensors enhance the perception range, making traffic participants behind the buses visible. In total, this ten-second-long sequence contains 2,790 labeled 3D objects during the daytime.

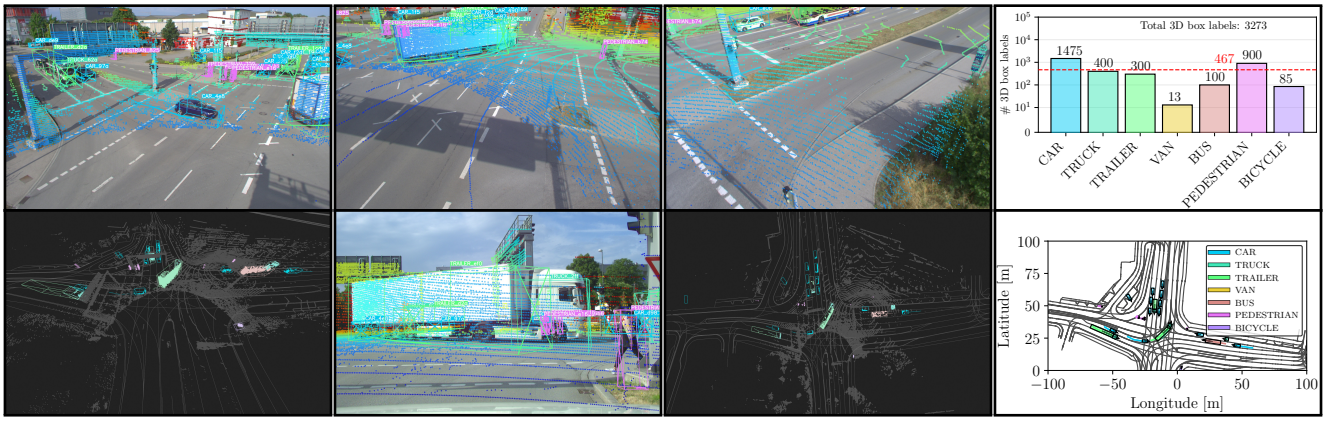


Figure 13. Visualization of **drive_12** of the *TUMTraf-V2X* dataset. This sequence with 3,273 3D boxes shows multiple occlusion scenarios. In one scenario a truck is occluding multiple pedestrians. The roadside sensors can perceive the objects behind the truck so that the ego vehicle becomes aware of them.

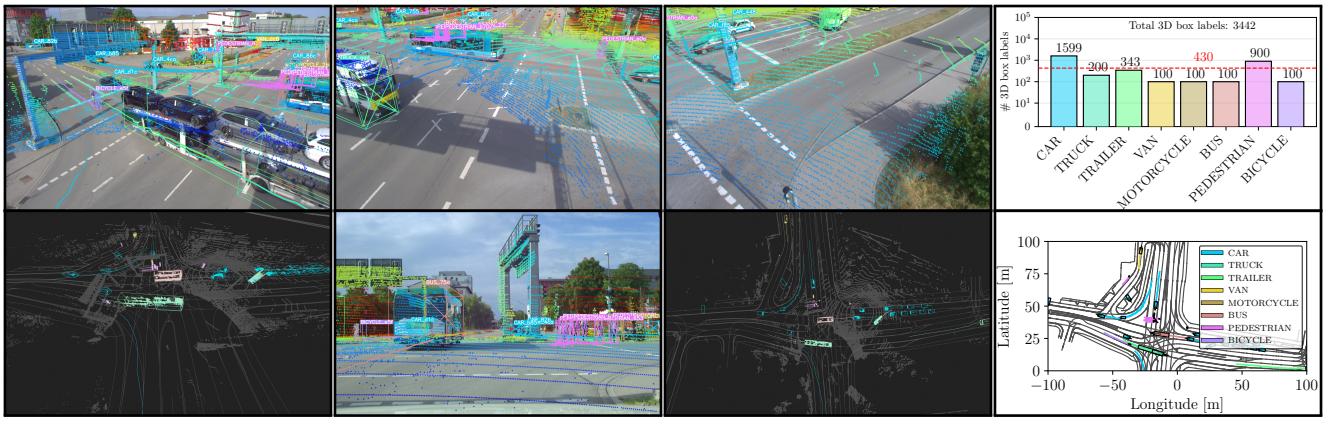


Figure 14. Visualization of **drive_15** of the *TUMTraf-V2X* dataset. In this drive, a bus is occluding a car which the roadside sensors can perceive. This is the largest sequence during daytime with 3,442 labeled 3D objects.

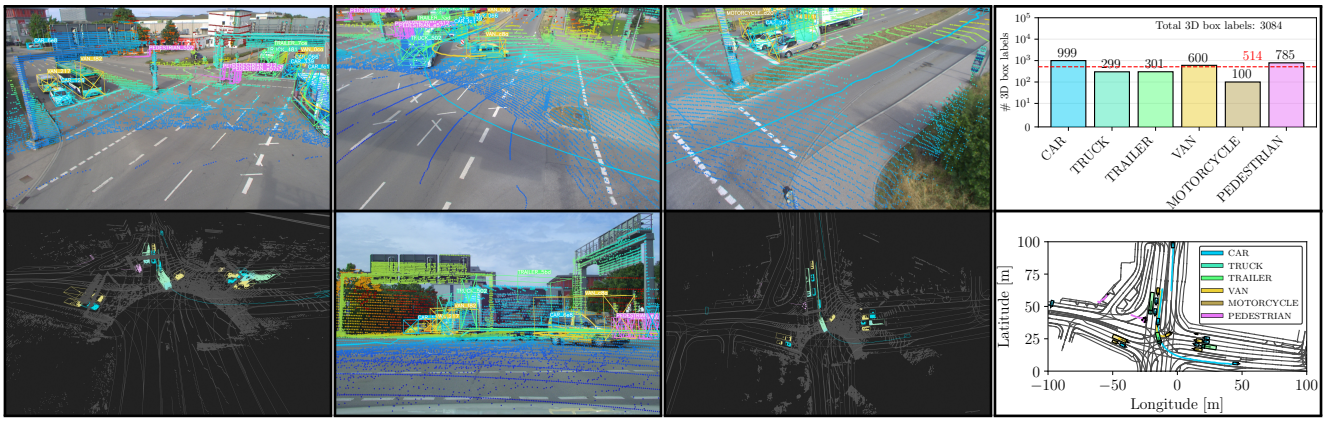


Figure 15. Visualization of **drive_22** of the *TUMTraf-V2X* dataset. In this drive, many vehicles are performing a U-turn maneuver and occlude some pedestrians waiting at a red traffic light. The pedestrians are within the field of view of the roadside sensors and can be perceived. This sequence contains 3,084 labeled 3D objects.

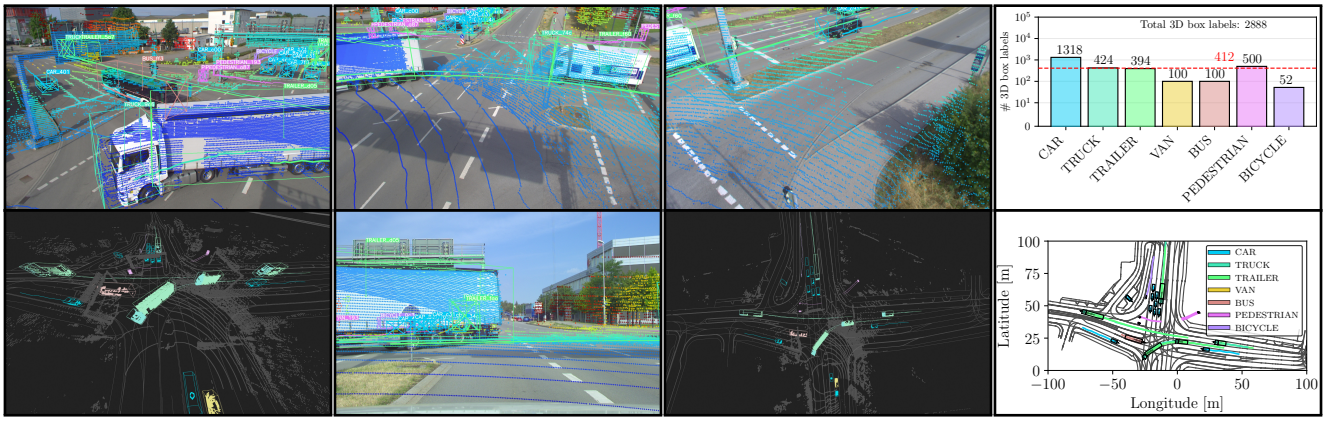


Figure 16. Visualization of **drive_26** of the *TUMTraf-V2X* dataset. In this drive, multiple trucks and trailers occlude traffic participants. These traffic participants are visible from the elevated roadside cameras and LiDAR mounted on the infrastructure. This sequence contains 2,888 labeled 3D objects.

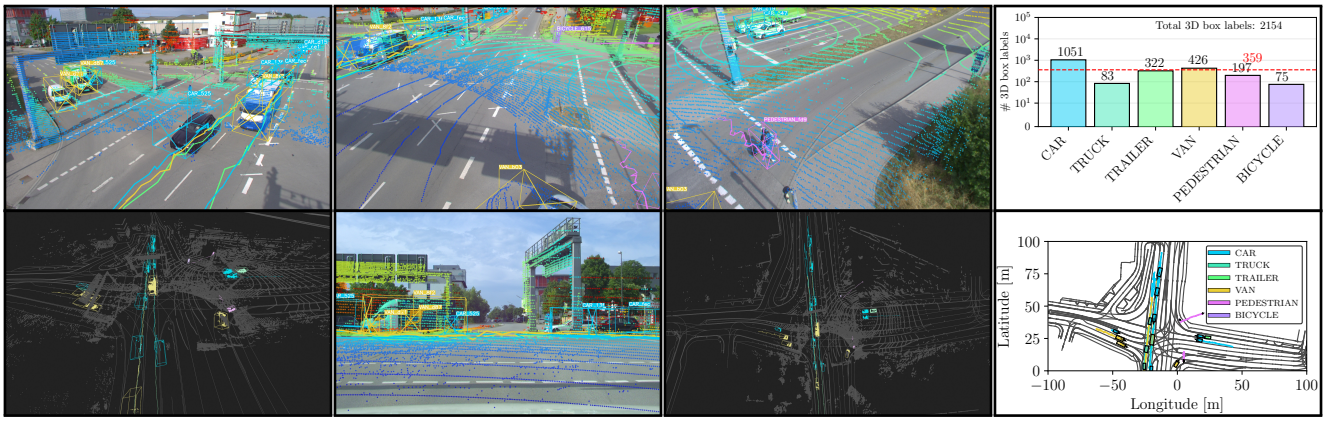


Figure 17. Visualization of **drive_33** of the *TUMTraf-V2X* dataset. In this scenario, a truck is occluding multiple objects that can be perceived by the roadside camera and LiDAR. Here, 2,154 3D objects were labeled.

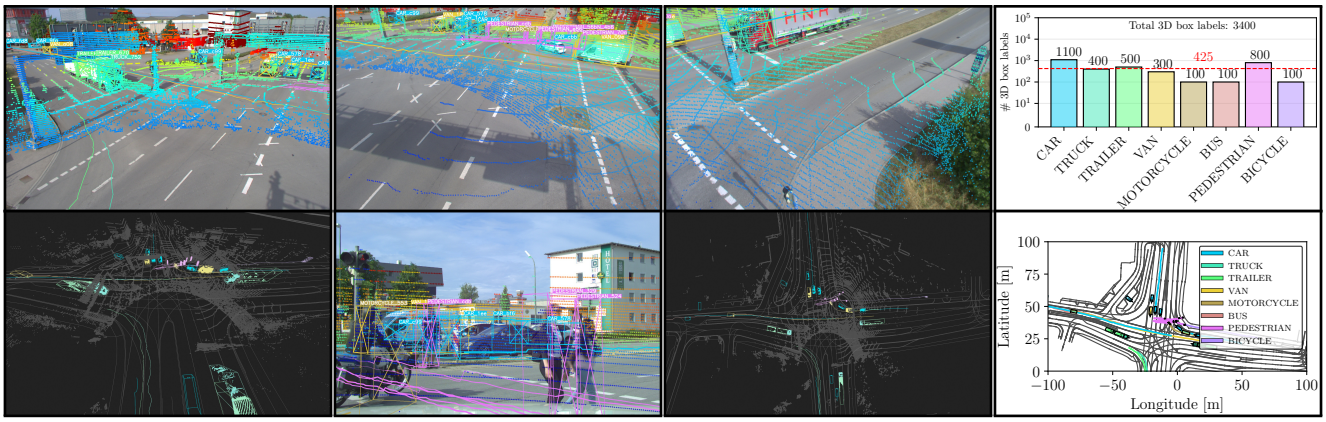


Figure 18. Visualization of **drive_41** of the *TUMTraf-V2X* dataset. In this example, a motorcyclist is overtaking the ego vehicle that gives way to pedestrians crossing the road. This sequence contains 3,400 labeled 3D objects.

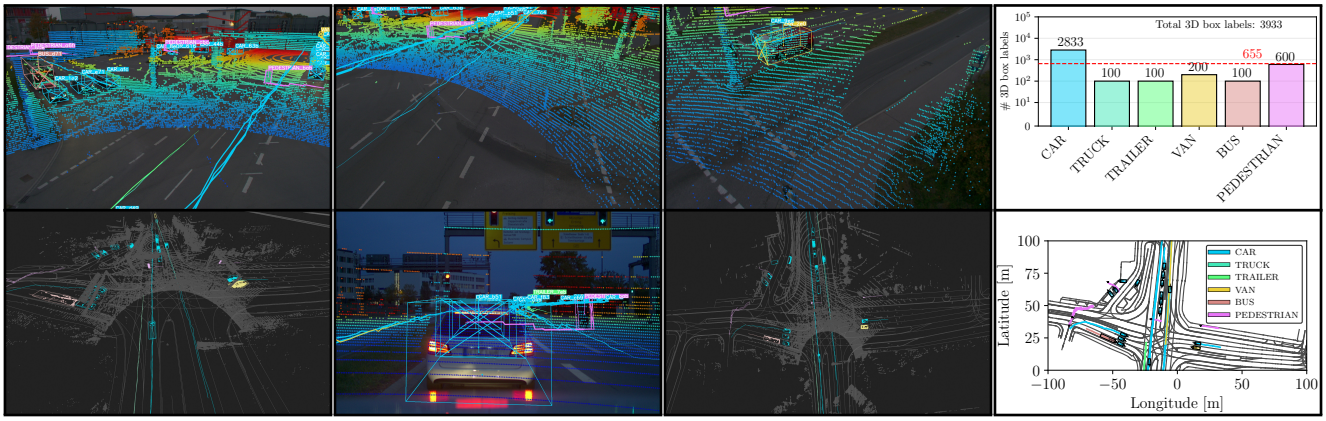


Figure 19. Visualization of **drive_42** of the *TUMTraf-V2X* dataset. This night scene contains a traffic violation and is the largest sequence in the dataset, with 3,933 3D objects. A pedestrian runs the red light after a fast-moving vehicle has crossed the intersection.

Table 8. Evaluation results (mAP_{BEV}) of the *CoopDet3D* and *CoopCMT* model on our *TUMTraf-V2X* test set in south2 FOV.

Config.		$\mathbf{mAP_{BEV}}\uparrow$	
Domain	**Modality**	CoopDet3D	CoopCMT
Vehicle	Camera	46.83	**81.21** (+34.38)
Vehicle	LiDAR	85.33	**86.88** (+1.55)
Infra.	Camera	61.98	**79.50** (+17.52)
Infra.	LiDAR	92.86	**93.18** (+0.32)
Infra.	Cam+LiDAR	92.92	**93.63** (+0.71)
Coop.	LiDAR	93.93	**94.27** (+0.38)

H.2. Multi-object tracking

Multiple Object Tracking Accuracy (MOTA) and *Multiple Object Tracking Precision* (MOTP) are the most widely used metrics to evaluate tracking performance. MOTA (Eq. 8) considers the main factors affecting tracking performance including *False Positives* (FP), *False Negatives* (FN), and *ID Switches* (IDS). GT_t is the number of ground truth objects at time t.

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDS_t)}{\sum_t GT_t} \tag{8}$$

MOTP (Eq. 9) is used to measure the precision of the tracked object's position, where d_t^i and c_t represent the distance between the predicted object and its actual position at time t and the number of matches at time t respectively.

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{9}$$

IDP and IDR are the ID precision and recall measuring the fraction of tracked detections that are correctly assigned to a unique ground truth ID. The IDF1 metric is the ratio of correctly identified tracked detections over the average number of ground truth objects (GT). The basic idea of IDF1 is to combine IDP and IDR into a single number. In addition, each trajectory can be classified as mostly tracked

(MT), partially tracked (PT), and mostly lost (ML). A target is mostly tracked if it is successfully tracked for at least 80% of its life span, mostly lost if it is successfully tracked for at most 20%. All other targets are partially tracked.

I . Further experiments

We extend our experiments to consider multiple FOVs, baseline models, and different tasks made possible through the proposed *TUMTraf-V2X* dataset.

I.1. CoopDet3D

Previously we discussed the performance of the proposed *CoopDet3D* model with *PointPillars* 512_2x and *YOLOv8* backbones in South2 camera FOV. In Table 7 we show the quantitative results of the same model in South1 camera FOV. Like the South2 camera FOV, we observe that the *CoopDet3D* cooperative model performs better than the vehicle-only perception model (+7.47 3D mAP). Fig. 21 shows qualitative results of *CoopDet3D* on drive_42.

I.2. CoopCMT

In addition to *CoopDet3D*, we build another cooperative fusion model: *CoopCMT* for benchmarking, based on cross-modal transformers (CMT) [53]. Similar to the proposed *CoopDet3D* model, the *CoopCMT* cooperative perception model uses separate vehicle and infrastructure backbones for feature extraction. Then, the extracted infrastructure and vehicle deep features are concatenated using a *MaxPooling* layer (similar to *PillarGrid* [3]), and finally passed onto the 3D detection head. Thus, this architecture is similar to the *CoopDet3D* architecture, where the *BEVFusion*-based backbones and head, are replaced with the corresponding counterpart from the *CMT* model. Note, that since transformer-based models require a large amount of data to be trained, the infrastructure backbone was first pre-trained on the *TUMTraf Intersection* dataset [64], and the vehicle backbone was pre-trained on the *nuScenes* dataset [8], to fit the domain. We compare the performance of the *CoopCMT* model with *CoopDet3D* in Table 8 and see that it outperforms the *CoopDet3D* model in all domains and modalities.

From Table 8, we observe a general trend in which the *CoopCMT* cooperative fusion model performs better in terms of the mAP_{BEV} compared to the *CoopDet3D* model. However, it must be noted that the *CoopCMT* model uses a transformer-based architecture, and as such, the model complexity is higher, resulting in slower inference time. For future research, the *CoopCMT* model will be studied further in terms of the model complexity and FPS to ensure that this model can perform in near real-time and can be deployed on edge devices.

I.3. 3D multi-object tracking

Next, we track the *CoopDet3D* detections in a post-processing step using two different trackers: *SORT* [6] and *PolyMOT* [29]. The quantitative evaluation results of 15 different metrics are listed in Table 6. We use a distance threshold of 5 m for the *SORT* tracker. The *PolyMOT* tracker performs best in all metrics except PT and MOTP. Qualitative results are shown in Fig. 11.

J . Statistics of all drives

Detailed statistics of all labeled sequences are seen in Figs. 12 to 19. The last driving sequence (drive_42) was recorded during nighttime and contains a traffic violation scenario in which a pedestrian is running the red light. All other sequences contain daytime traffic with heavy occlusion scenarios. We split our dataset into a training (80%), validation (10%), and test (10%) set using stratified sampling to get a well-balanced split. The distribution of object classes of our training, validation, and test set is shown in Fig. 20.

K . Detailed dataset visualization

We provide detailed dataset visualizations for different challenging traffic scenarios at an urban intersection, including tailgating, overtaking, U-turns, traffic violations, and occlusion scenarios. In one scene, a pedestrian runs a red light after a vehicle is crossing. We show each scenario's surround-view images, BEV projections on an HD map, point cloud visualizations, and a class distribution plot. Visualization videos for all labeled sequences are provided on our website: https://tum-traffic-dataset.github.io/tumtraf-v2x.

L . Failure cases and limitations

Failure cases are essential to understand the weakness of our dataset and model and to provide some guidance for future work. Note that, for brevity, we do not consider the network communication latency between the sensors.
We have tested our *CoopDet3D* model in day and night scenarios in different weather conditions. Some future work will include further tests under harsh weather conditions such as heavy rain, snow, and fog. Apart from object detection, cooperative perception poses many other challenges due to the asynchrony between the vehicle and infrastructure sensors, and the transmission delay further exacerbates this issue. While the suggested model may not fully account for these considerations, it is recommended that future research focuses on addressing these challenges through extensive live tests.

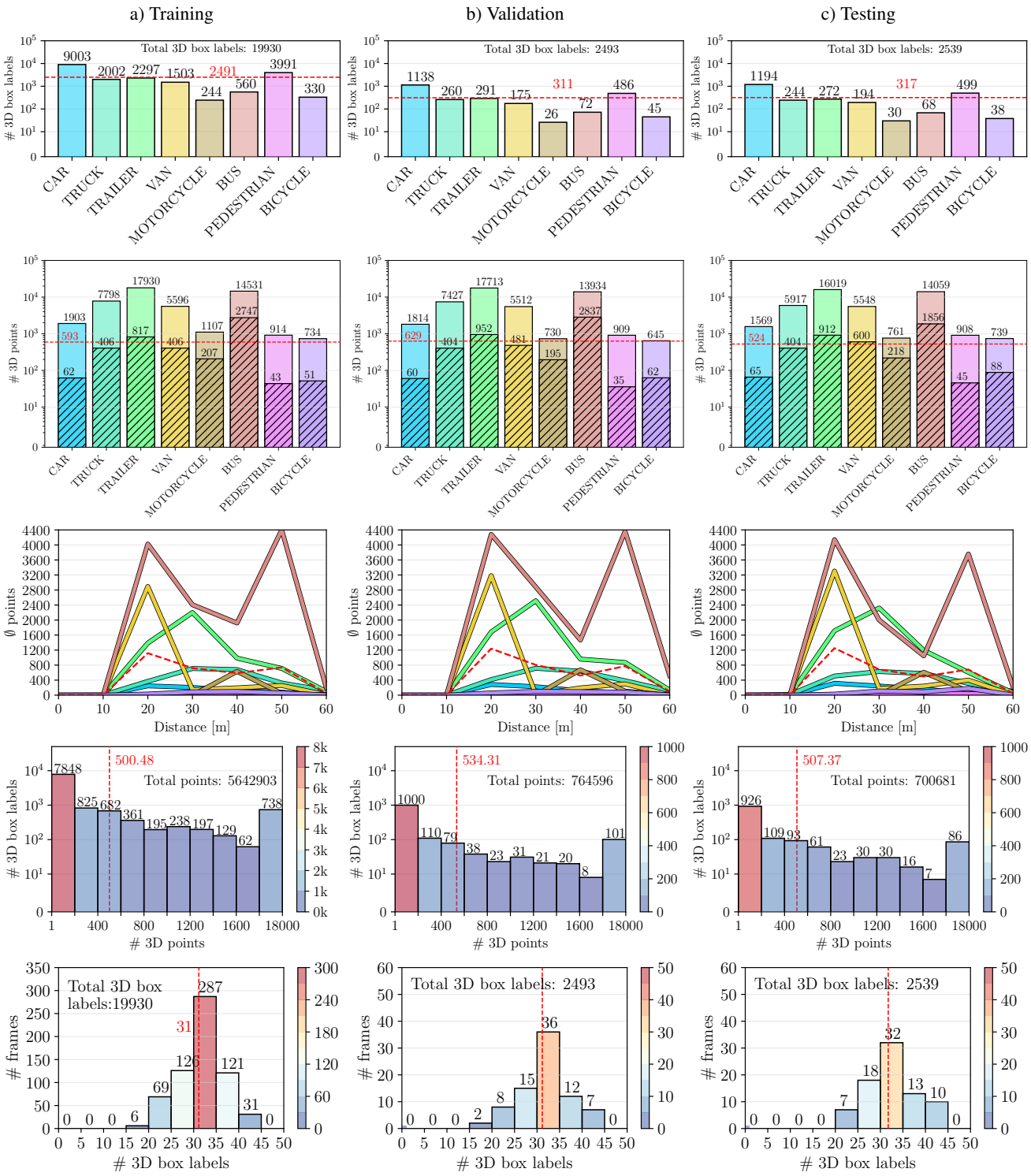


Figure 20. Distribution of our *TUMTraf-V2X* dataset (version 1.0) into a) training, b) validation, and c) test set. From top to bottom: We show the distribution of object classes within each set with the average number of 3D box labels marked in red, the distribution of 3D points for each category and each set, the labeled distance and class density for each object class and set, a histogram of 3D box densities for each set, and a histogram of frame densities for each set.

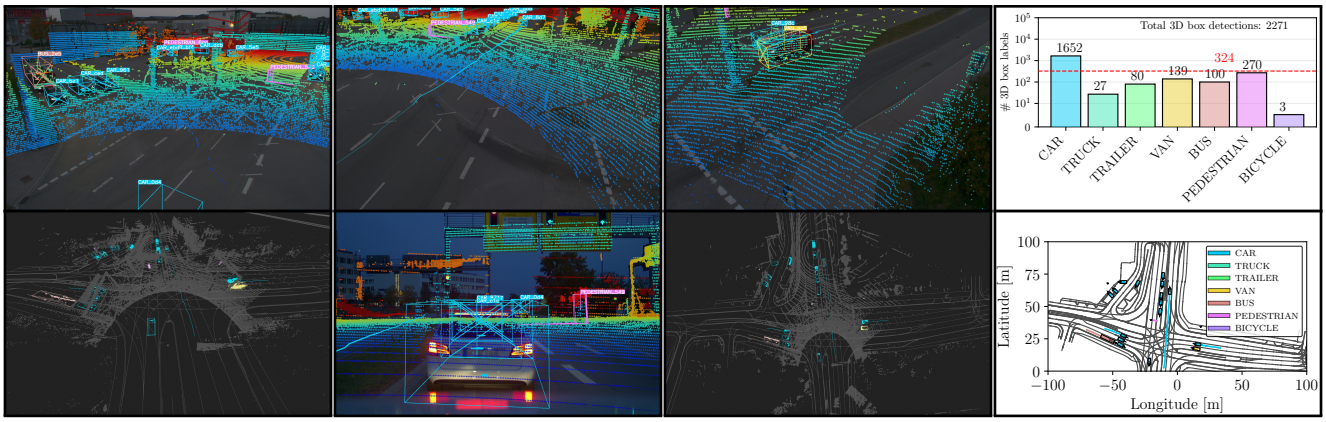


Figure 21. Qualitative results of *CoopDet3D* on **drive_42** of our *TUMTraf-V2X* dataset of a night scene. We project the detections into point cloud scans and camera images. Moreover, we visualize object tracks in a bird's-eye view and an HD map. Finally, we show the distribution of detections in a bar chart.

References

[1] Hasan Asy'ari Arief, Mansur Arief, Guilin Zhang, Zuxin Liu, Manoj Bhat, Ulf Geir Indahl, Håvard Tveite, and Ding Zhao. Sane: smart annotation and evaluation tools for point cloud data. *IEEE Access*, 8:131848–131858, 2020. 3

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers, 2022. 3, 4

[3] Zhengwei Bai, Guoyuan Wu, Matthew J Barth, Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi. Pillargrid: Deep learning-based cooperative perception for 3d object detection from onboard-roadside lidar. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1743–1749. IEEE, 2022. 4, 7, 9

[4] Zhengwei Bai, Guoyuan Wu, Matthew J Barth, Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi. Vinet: Lightweight, scalable, and heterogeneous cooperative perception for 3d object detection. *Mechanical Systems and Signal Processing*, 204:110723, 2023. 4

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 4

[6] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 4, 9

[7] Steffen Busch, Christian Koetsier, Jeldrik Axmann, and Claus Brenner. Lumpi: The leibniz university multi-perspective intersection dataset. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1127–1134. IEEE, 2022. 3

[8] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 4, 9

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 3, 7, 5

[10] Wesley Chen, Andrew Edgley, Raunak Hota, Joshua Liu, Ezra Schwartz, Aminah Yizar, Neehar Peri, and James Purtilo. Rebound: An open-source 3d bounding box annotation tool for active learning. *AutomationXP @ CHI 2023*, 2023. 3

[11] Ziming Chen, Yifeng Shi, and Jinrang Jia. Transiff: An instance-level feature fusion framework for vehicle-infrastructure cooperative 3d detection with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18205–18214, 2023. 4

[12] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 4

[13] MMYOLO Contributors. MMYOLO: OpenMMLab YOLO series toolbox and benchmark. https://github.com/open-mmlab/mmyolo, 2022. 8

[14] Christian Creß, Zhenshan Bing, and Alois C. Knoll. Intelligent transportation systems using roadside infrastructure: A literature survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–0, 2023. 2

[15] Christian Creß, Walter Zimmer, Leah Strand, Maximilian Fortkord, Siyi Dai, Venkatnarayanan Lakshminarasimhan, and Alois Knoll. A9-dataset: Multi-sensor infrastructure-based dataset for mobility research. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 965–970, 2022. 3

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3

[17] Achref Doula, Tobias Güdelhöfer, Andrii Matviienko, Max Mühlhäuser, and Alejandro Sanchez Guinea. Pointcloudlab: An environment for 3d point cloud annotation with adapted

visual aids and levels of immersion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11640–11646, 2023. 3

[18] LF AI & Data Foundation. Xtreme1 - the next gen platform for multisensory training data, 2023. Software available from https://github.com/xtreme1-io/xtreme1/. 3

[19] 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, 3

[20] Nicco Hagedorn. OpenLABEL Concept Paper. 2, 5

[21] Yuze He, Li Ma, Zhehao Jiang, Yi Tang, and Guoliang Xing. Vi-eye: Semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 573–586, 2021. 4

[22] Yue Hu, Yifan Lu, Runsheng Xu, Weidi Xie, Siheng Chen, and Yanfeng Wang. Collaboration helps camera overtake lidar in 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2023. 4

[23] Glenn Jocher, K Nishimura, T Mineeva, and R Vilariño. yolov5. *Code repository*, 2020. 2

[24] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo, 2023. 8

[25] Annkathrin Krämmer, Christoph Schöller, Dhiraj Gulati, Venkatnarayanan Lakshminarasimhan, Franz Kurz, Dominik Rosenbaum, Claus Lenz, and Alois Knoll. Providentia-a large-scale sensor system for the assistance of autonomous vehicles and its evaluation. *Journal of Field Robotics*, pages 1156–1176, 2022. 2

[26] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. Deviant: Depth equivariant network for monocular 3d object detection. In *European Conference on Computer Vision*, pages 664–683. Springer, 2022. 4

[27] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 4, 8

[28] E Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. Sustech points: A portable 3d point cloud interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1108–1115, 2020. 2, 3

[29] Xiaoyu Li, Tao Xie, Dedong Liu, Jinghan Gao, Kun Dai, Zhiqiang Jiang, Lijun Zhao, and Ke Wang. Poly-mot: A polyhedral framework for 3d multi-object tracking. *arXiv preprint arXiv:2307.16675*, 2023. 4, 9

[30] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921, 2022. 2, 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 8

[32] 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. 5

[33] 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. 8

[34] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023. 4, 7

[35] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. 3

[36] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. 2004. 4

[37] Donghao Qiao and Farhana Zulkernine. Cobevfusion: Cooperative perception with lidar-camera bird's-eye view fusion. *arXiv preprint arXiv:2310.06008*, 2023. 4

[38] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, page 5. Kobe, Japan, 2009. 5

[39] Christoph Sager, Patrick Zschech, and Niklas Kühl. labelcloud: A lightweight domain-independent labeling tool for 3d object detection in point clouds, 2021. 3

[40] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 4

[41] Rui Song, Chenwei Liang, Hu Cao, Zhiran Yan, Walter Zimmer, Markus Gross, Andreas Festag, and Alois Knoll. Collaborative semantic occupancy prediction with hybrid feature fusion in connected automated vehicles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024. 4

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 3

[43] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. TorchSparse++: Efficient Point Cloud Engine. In *Computer Vision and Pattern Recognition Workshops CVPRW*, 2023. 8

[44] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 4

[45] Bernie Wang, Virginia Wu, Bichen Wu, and Kurt Keutzer. Latte: accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 265–272. IEEE, 2019. 3

[46] Sizhe Wei, Yuxi Wei, Yue Hu, Yifan Lu, Yiqi Zhong, Siheng Chen, and Ya Zhang. Robust asynchronous collaborative 3d detection via bird's eye view flow. *arXiv preprint arXiv:2309.16940*, 2023. 4

[47] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 3

[48] Zizhang Wu, Yuanzhu Gan, Lei Wang, Guilian Chen, and Jian Pu. Monopgc: Monocular 3d object detection with pixel geometry contexts. *preprint arXiv:2302.10549*, 2023. 4

[49] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pages 107–124. Springer, 2022. 2, 3, 4

[50] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 3

[51] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022. 2, 3

[52] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13712–13722, 2023. 2, 3

[53] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18268–18278, 2023. 4, 9

[54] 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. 4

[55] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21341–21350, 2022. 3, 4

[56] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 5

[57] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022. 2, 3, 4

[58] Haibao Yu, Yingjuan Tang, Enze Xie, Jilei Mao, Jirui Yuan, Ping Luo, and Zaiqing Nie. Vehicle-infrastructure cooperative 3d object detection via feature flow prediction. *arXiv preprint arXiv:2303.10552*, 2023. 4

[59] 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, 3

[60] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 5

[61] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 8

[62] Walter Zimmer, Akshay Rangesh, and Mohan Trivedi. 3d bat: A semi-automatic, web-based 3d annotation toolbox for full-surround, multi-modal data streams. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1816–1821. IEEE, 2019. 5, 2, 3

[63] Walter Zimmer, Joseph Birkner, Marcel Brucker, Huu Tung Nguyen, Stefan Petrovski, Bohan Wang, and Alois C. Knoll. Infradet3d: Multi-modal 3d object detection based on roadside infrastructure camera and lidar sensors. In *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023. 4, 5, 8

[64] Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois C Knoll. Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception. In *2023 IEEE Intelligent Transportation Systems ITSC*. IEEE, 2023. 3, 5, 8, 9

[65] Walter Zimmer, Jialong Wu, Xingcheng Zhou, and Alois C Knoll. Real-time and robust 3d object detection with roadside lidars. In *Proceedings of the 12th International Scientific Conference on Mobility and Transport: Mobility Innovations for Growing Megacities*, pages 199–219. Springer, 2023. 4