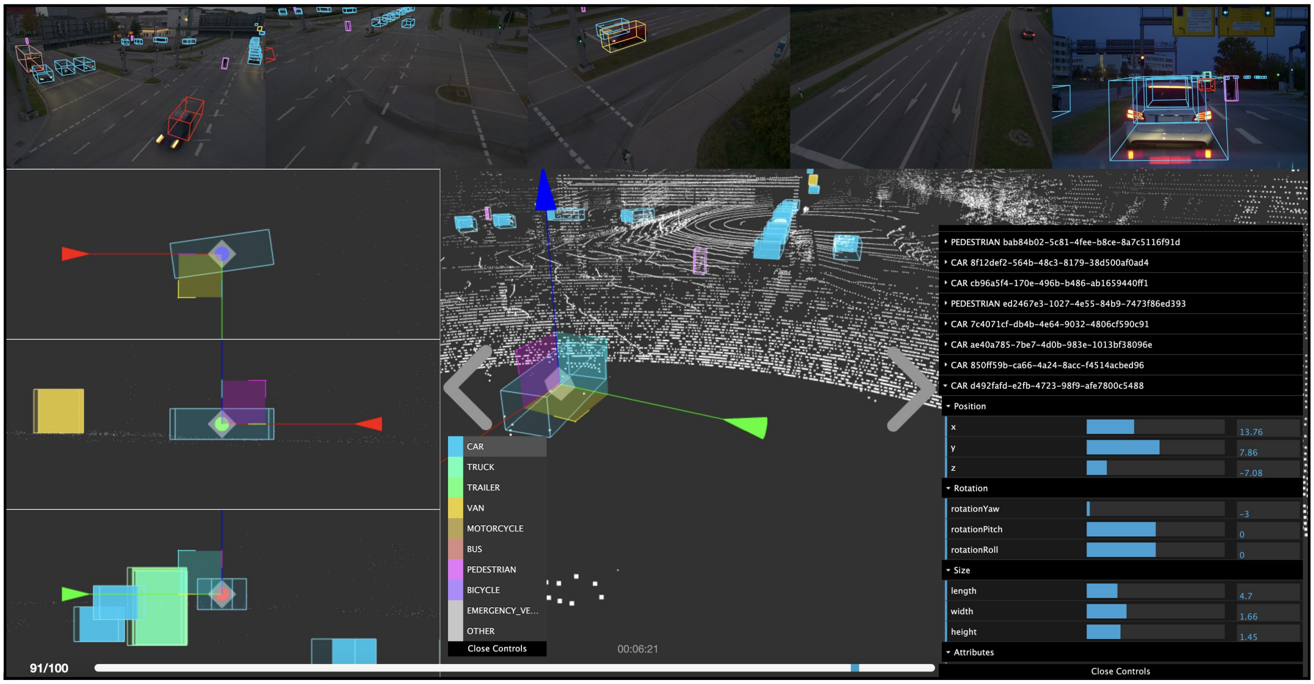
TUMTraf V2X Cooperative Perception Dataset

Supplementary Material

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



- PEDESTRIAN bab84b02-5c81-4fee-b8ce-8a7c5116f91d
- CAR 8f12def2-564b-48c3-8179-38d500af0ad4
- CAR cb96a5f4-170e-496b-b486-ab1659440ff1
- PEDESTRIAN ed2467e3-1027-4e55-84b9-7473f86ed393
- CAR 7c4071cf-db4b-4e64-9032-4806cf590c91
- CAR ae40a785-7be7-4d0b-983e-1013bf38096e
- CAR 850ff59b-ca66-4a24-8acc-f4514acbed96
- CAR d492fafd-e2fb-4723-98f9-afe7800c5488

Position

x	13.76
y	7.86
z	-7.08

Rotation

rotationYaw	-3
rotationPitch	0
rotationRoll	0

Size

length	4.7
width	1.66
height	1.45

Attributes

Figure 1. 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:

$$\min \mathop{\mathbb{E}}_{y_j \in Y(t)} \left[\min_{\tilde{y}_k \in \tilde{Y}(t)} d_\theta \left(y_j, \tilde{y}_k \right) \right] \tag{1}$$

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 [10] 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 [27] 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 [11] 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 1. Comparison of the proposed 3D BAT v24.3.2 annotation tool with other state-of-the-art 3D annotation tools.
🟢 Feature provided or is outstanding 🟡 Feature unknown or is less important 🔴 Feature not provided or is limited

Tool	3D BAT [27]	LATTE [16]	SAnE [1]	SUSTech POINTS[11]	Label Cloud[14]	ReBound [6]	PointCloud Lab[8]	Xtreme1 [9]	**3D BAT (Ours)***
Year	2019	2020	2020	2020	2021	2023	2023	2023	2024
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	580	374	62	670	461	20	-	542	580
Citations	58	36	16	33	16	0	2	0	58
License	Custom	Apach. 2.0	Apach. 2.0	GPL 3.0	GPL 3.0	Apach. 2.0	-	Apach. 2.0	Custom

*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 [14] 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 [6] is an open source 3D bounding box annotation tool designed to utilize active learning. It supports loading, visualizing, and extending existing datasets like nuScenes [5], Waymo [15] or Argoverse 2.0 [17]. Model predictions can be analyzed and corrected in a 3D view and exported to specific formats.

PointCloudLab [8] 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* [9] 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 [19] is an open cooperative detection framework for autonomous driving which supports popular simulated datasets such as OPV2V [20] and V2XSet [18]. 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

Table 2. 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* [4]	36.313	21.029	26.634	43.235	74.657	3400	5	**18**	11	499	1920	439	110	15.647	**100.185**
PolyMOT [12]	**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.

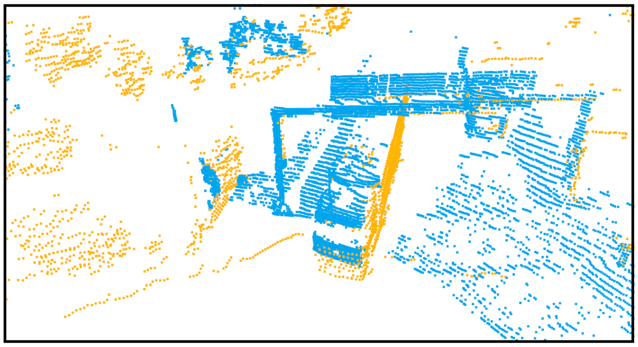


Figure 2. Point cloud registration results of an onboard LiDAR point cloud (orange) and a roadside LiDAR point cloud (blue).

used for visualization. V2V4Real [21] extends the OpenCOOD development kit, to support real-world data and additional perception tasks. Furthermore, data augmentation is also an additional feature that can be enabled when training the model.

Furthermore, the DAIR V2X [25], 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 [5] and Rope3D devkit [23], 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 [7].

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. 2.

$$T_{VI}^{0} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

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}.

$$P_{VI} = P_I \oplus (P_V \cdot T_{VI}) \tag{3}$$

Fig. 2 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

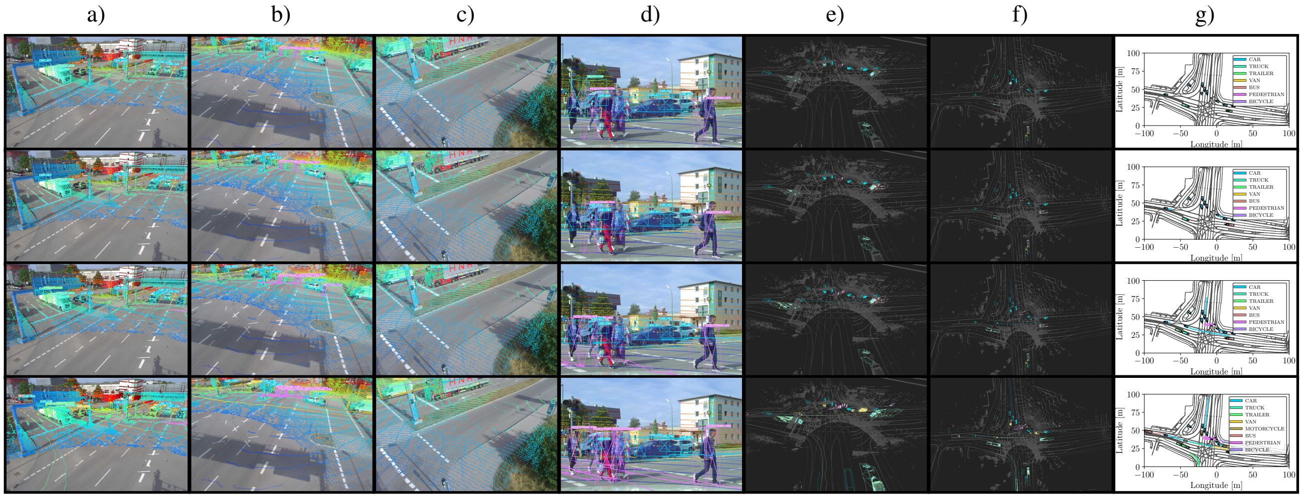


Figure 3. Tracking results on **drive_42** test sequence of the *TUMTraf-V2X* dataset. From top to bottom: *CoopDet3D* detections, *CoopDet3D* detections tracked by *SORT*, *CoopDet3D* detections tracked by *PolyMOT*, ground truth. a-c) Tracking results projected into roadside camera images. d) Tracking 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.

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 entropy loss L_{cls}, a weighted L_1 loss for the 3D box regression $\mathcal{L}_{reg}$, and a weighted IoU loss $\mathcal{L}_{IoU}$ [26] (see Eq. 4).

$$C_{match} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{reg} + \lambda_3 \mathcal{L}_{IoU}, \tag{4}$$

where: λ_1, λ_2, λ_3 are the coefficients for the individual cost terms. Given all matched pairs, a focal loss [13] is computed for the final classification. A penalty-reduced focal loss [24] 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.

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

Config.		**mAP$_{\text{BEV}}$ ↑**	**mAP$_{\text{3D}}$ ↑**			
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

H.1 . 3D Object Detection

As the most commonly utilized metric in 3D object detection tasks, *mean Average Precision* (mAP) (Eq. 5) 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* [5], 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.

$$mAP = \frac{1}{|\mathcal{C}||\mathcal{D}|} \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} AP_{c,d} \tag{5}$$

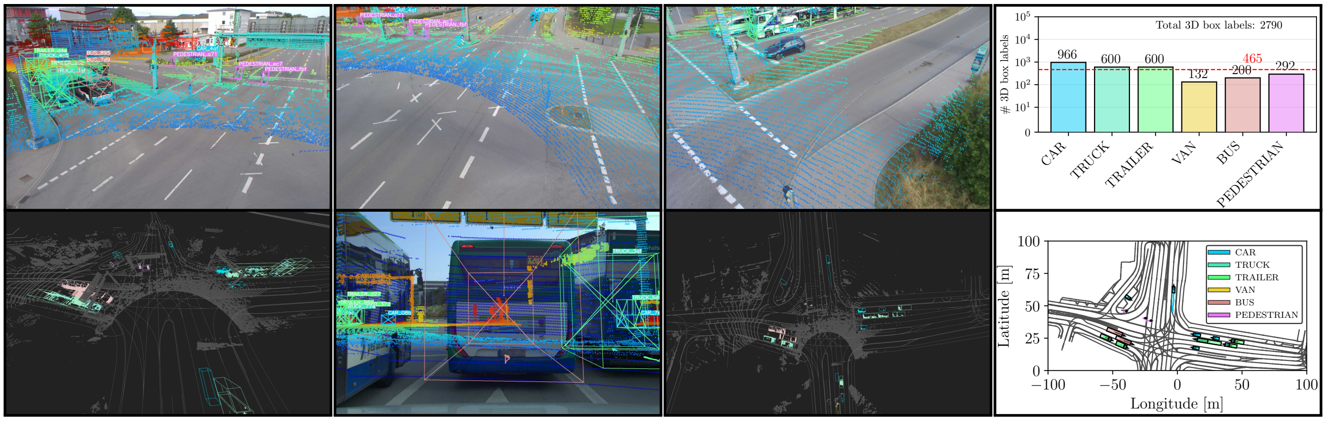


Figure 4. Visualization of **drive_07** of the *TUMTraf-V2X* dataset. In this example, the ego vehicle is occluded by two busses and two large trucks. The roadside sensors enhance the perception range, making traffic participants behind the buses visible. In total, this ten-second-long sequence contains 2,790 labeled 3D objects during the daytime.

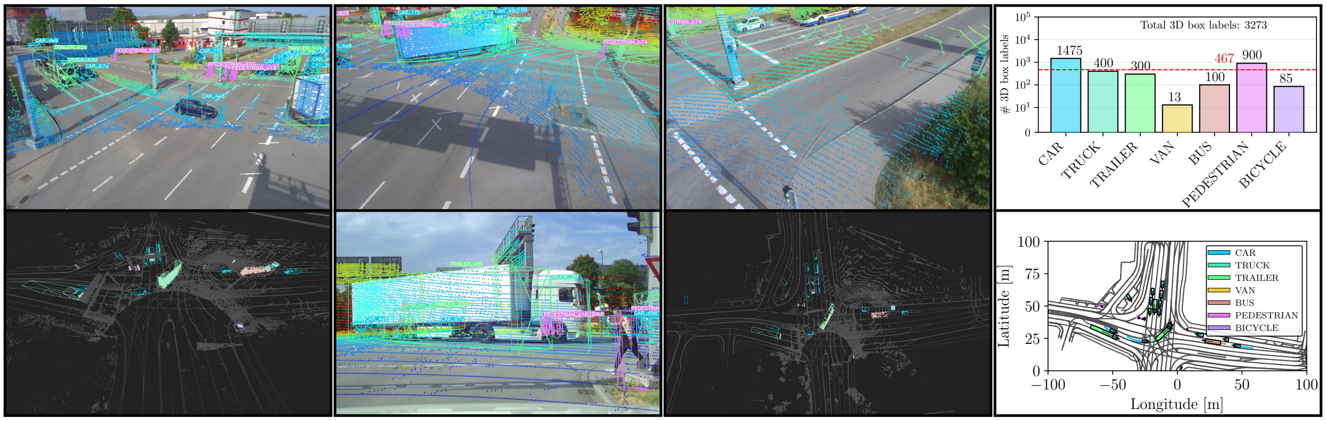


Figure 5. Visualization of **drive_12** of the *TUMTraf-V2X* dataset. This sequence with 3,273 3D boxes shows multiple occlusion scenarios. In one scenario a truck is occluding multiple pedestrians. The roadside sensors can perceive the objects behind the truck so that the ego vehicle becomes aware of them.

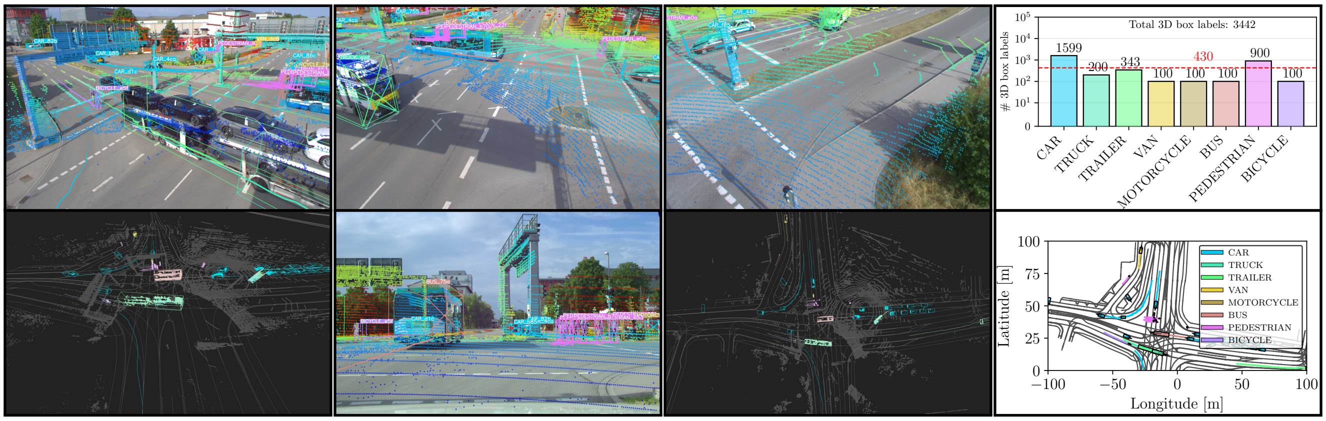


Figure 6. Visualization of **drive_15** of the *TUMTraf-V2X* dataset. In this drive, a bus is occluding a car which the roadside sensors can perceive. This is the largest sequence during daytime with 3,442 labeled 3D objects.

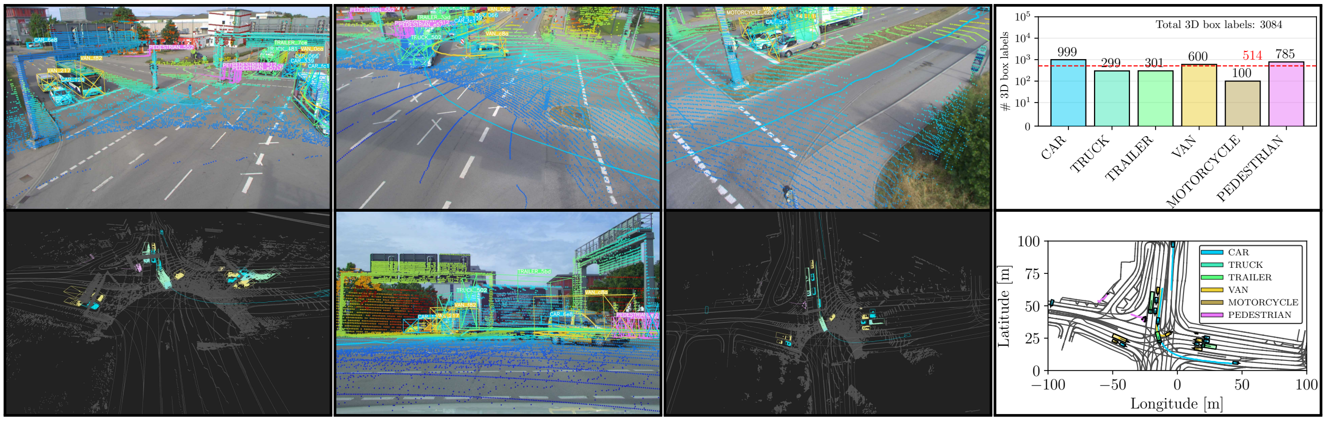


Figure 7. Visualization of **drive_22** of the *TUMTraf-V2X* dataset. In this drive, many vehicles are performing a U-turn maneuver and occlude some pedestrians waiting at a red traffic light. The pedestrians are within the field of view of the roadside sensors and can be perceived. This sequence contains 3,084 labeled 3D objects.

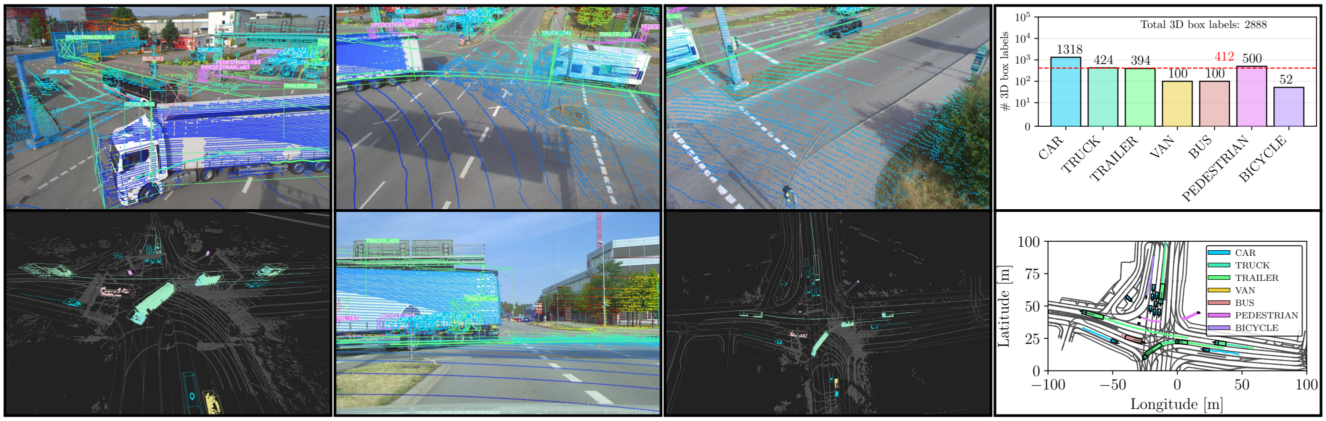


Figure 8. Visualization of **drive_26** of the *TUMTraf-V2X* dataset. In this drive, multiple trucks and trailers occlude traffic participants. These traffic participants are visible from the elevated roadside cameras and LiDAR mounted on the infrastructure. This sequence contains 2,888 labeled 3D objects.

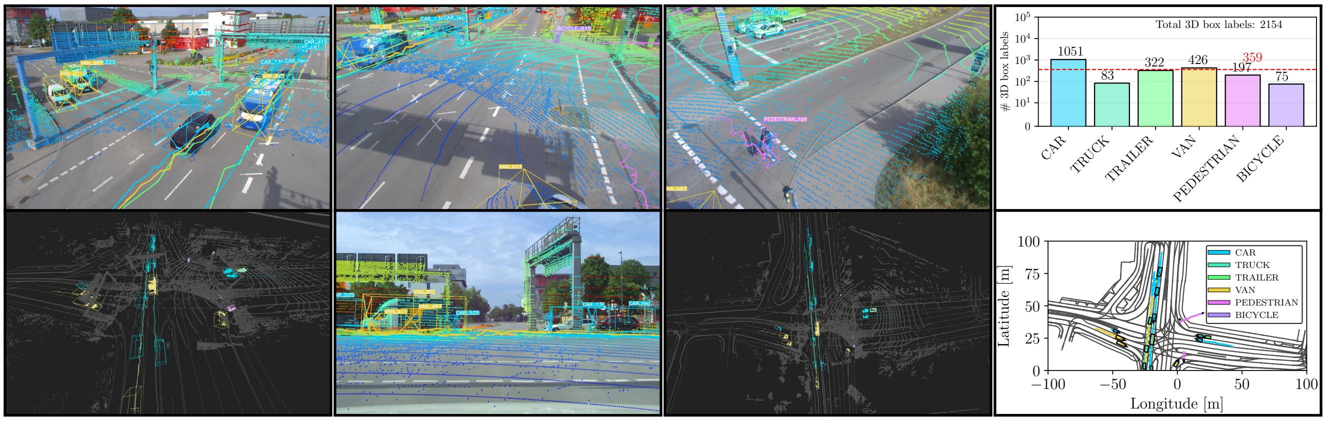


Figure 9. Visualization of **drive_33** of the *TUMTraf-V2X* dataset. In this scenario, a truck is occluding multiple objects that can be perceived by the roadside camera and LiDAR. Here, 2,154 3D objects were labeled.

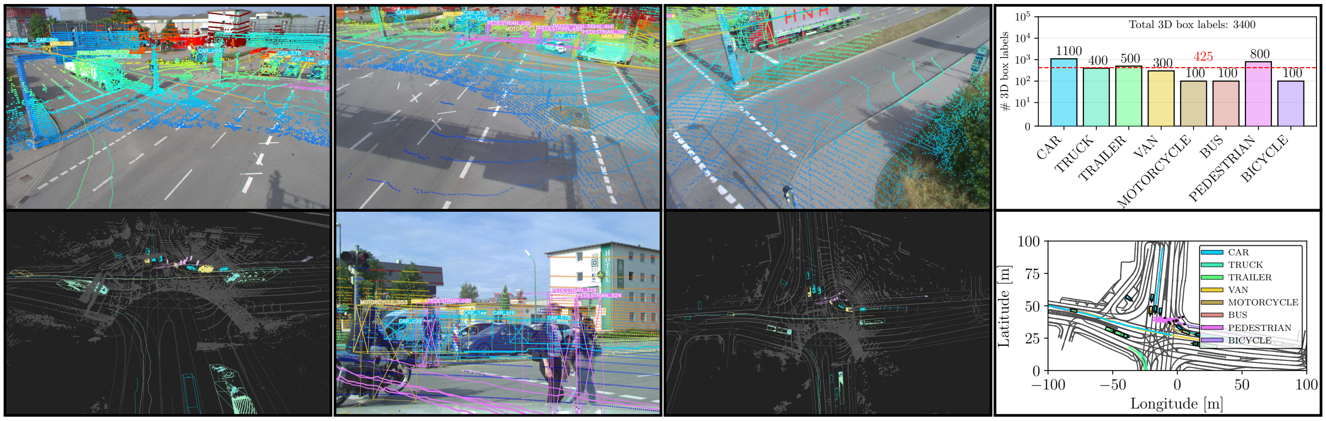


Figure 10. Visualization of **drive_41** of the *TUMTraf-V2X* dataset. In this example, a motorcyclist is overtaking the ego vehicle that gives way to pedestrians crossing the road. This sequence contains 3,400 labeled 3D objects and eight different object categories. Cars and pedestrians are highly represented, with 1,100 and 800 instances.

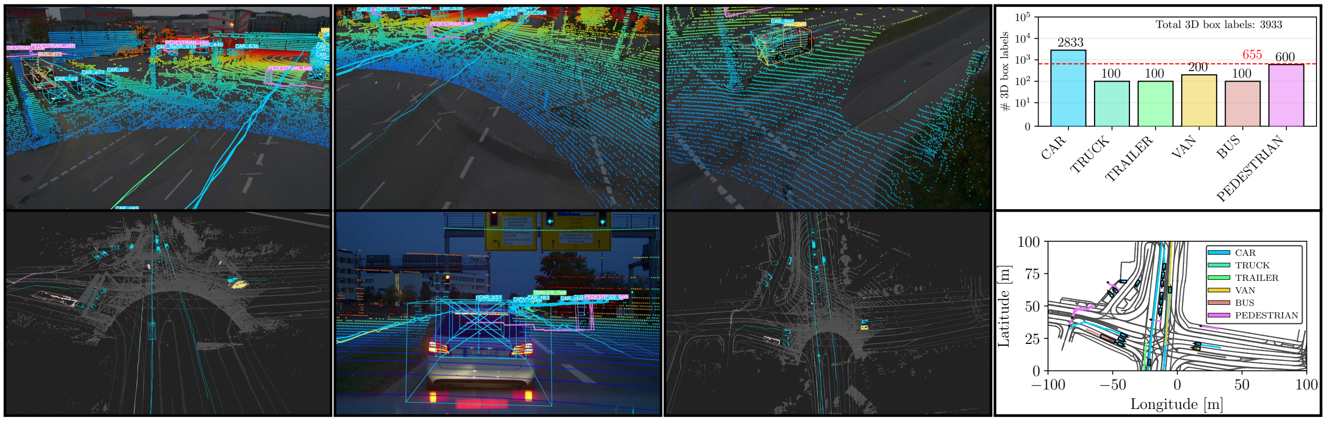


Figure 11. Visualization of **drive_42** of the *TUMTraf-V2X* dataset. This night scene contains a traffic violation and is the largest sequence in the dataset, with 3,933 3D objects. A pedestrian runs the red light after a fast-moving vehicle has crossed the intersection. This sequence contains six labeled object classes with 2,833 cars and 600 pedestrians.

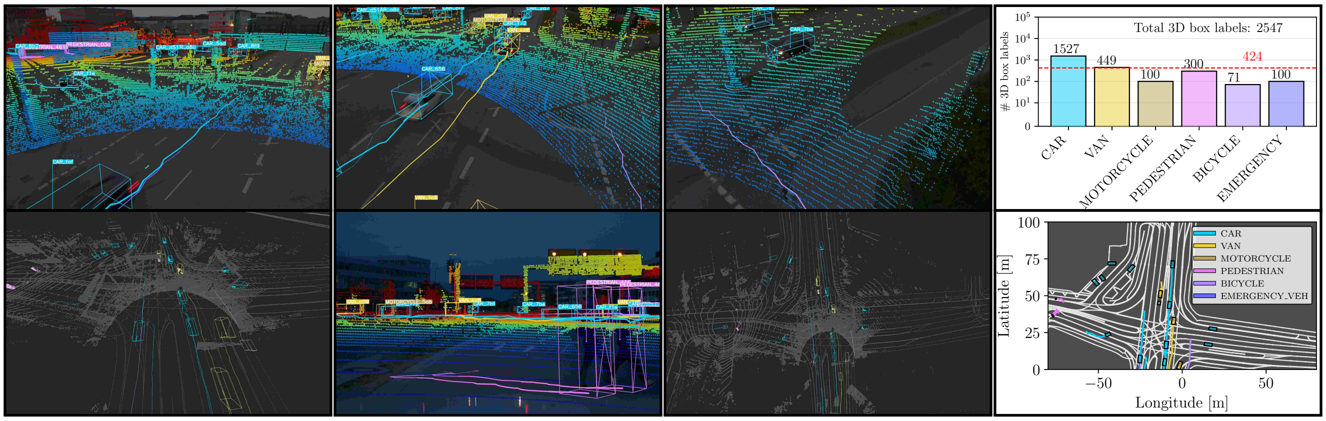


Figure 12. Visualization of **drive_44** of the *TUMTraf-V2X* dataset. This night scene shows a scenario, in that the ego vehicle is braking to avoid two pedestrians crossing the street in front of it. This scene contains 2,547 labeled 3D objects and is the only sequence that contains 100 labeled emergency vehicles.

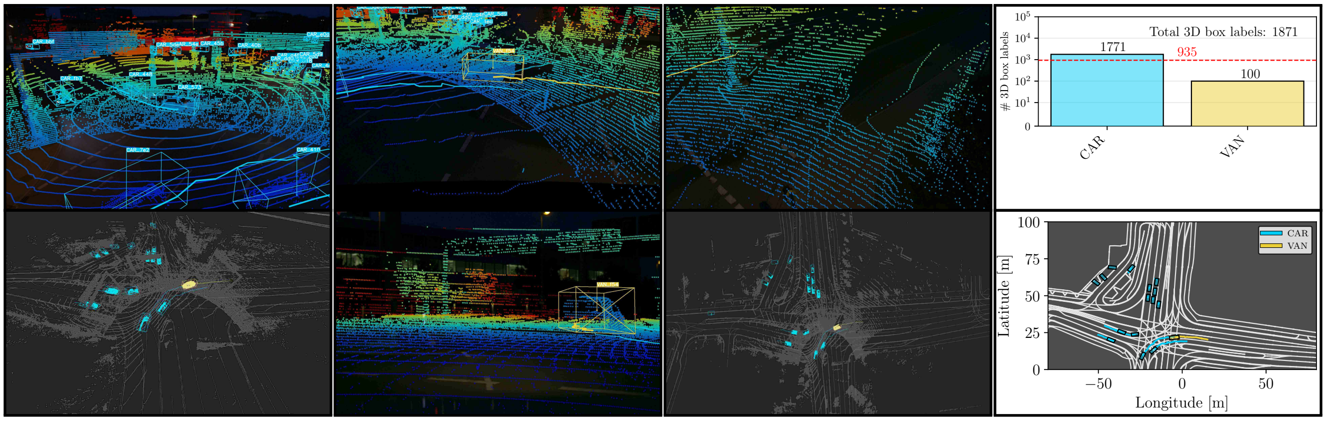


Figure 13. Visualization of **drive_53** of the *TUMTraf-V2X* dataset. This night scene shows a scenario, in that the ego vehicle is performing a U-turn. This scene contains 1,871 labeled 3D objects.

Table 4. Evaluation results (mAP_{BEV}) of the *CoopDet3D* and *CoopCMT* model on our *TUMTraf-V2X* test set in south2 FOV.

Config.		**mAP$_{\mathbf{BEV}}$ ↑**	
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. 6) 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.

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDS_t)}{\sum_t GT_t} \tag{6}$$

MOTP (Eq. 7) 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.

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{7}$$

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 3 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. 15 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) [22]. 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 *Max-Pooling* 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

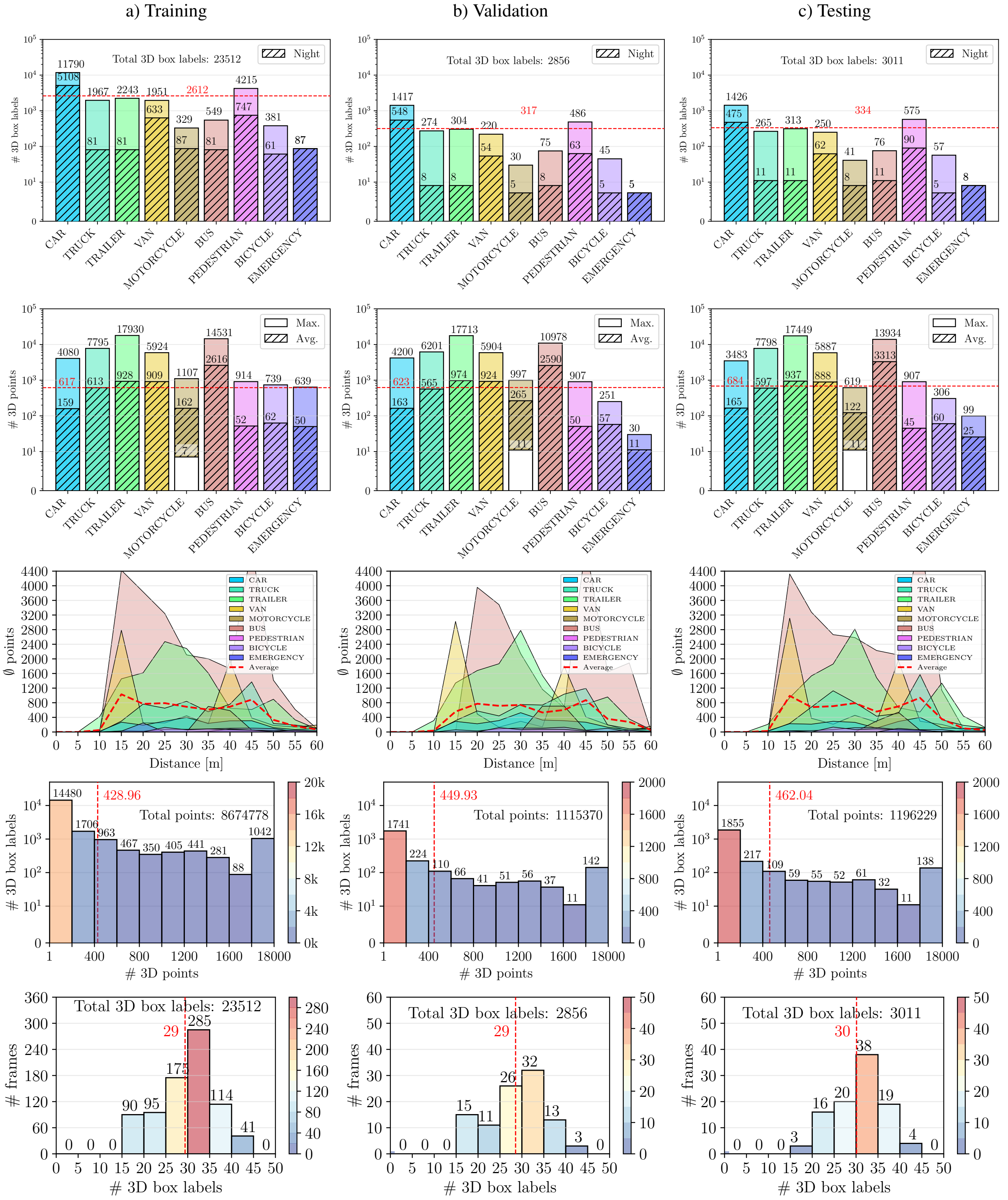


Figure 14. Distribution of our *TUMTraf-V2X* dataset into a) training, b) validation, and c) test set. From top to bottom: We show the distribution of object classes within each set with the average number of 3D box labels marked in red, the distribution of 3D points for each category and each set, the 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.

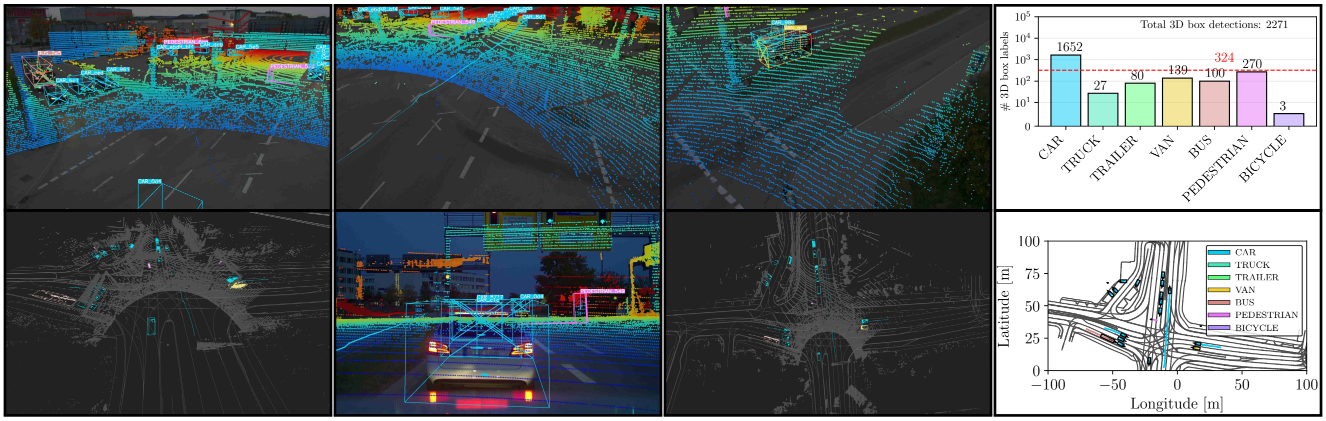


Figure 15. Qualitative results of *CoopDet3D* on **drive_42** of our *TUMTraf-V2X* dataset of a night scene. We project the detections into point cloud scans and camera images. Moreover, we visualize object tracks in a bird's-eye view and an HD map. Finally, we show the distribution of detections in a bar chart.

on the *TUMTraf Intersection* dataset [28], and the vehicle backbone was pre-trained on the *nuScenes* dataset [5], to fit the domain. We compare the performance of the *CoopCMT* model with *CoopDet3D* in Table 4 and see that it outperforms the *CoopDet3D* model in all domains and modalities.

From Table 4, 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* [4] and *PolyMOT* [12]. The quantitative evaluation results of 15 different metrics are listed in Table 2. 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. 3.

J . Statistics of all drives

Detailed statistics of all labeled sequences are seen in Figs. 4 to 10 and 13. 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. 14.

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.

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. 5

[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 *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1743–1749. IEEE, 2022. 9

[4] 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, 11

[5] 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 (CVPR)*, pages 11621–11631, 2020. 3, 4, 5, 11

[6] 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

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

[8] 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

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

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

[11] 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

[12] 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, 11

[13] 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

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

[15] 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. 3

[16] 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

[17] 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

[18] 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. 3

[19] 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

[20] 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. 3

[21] 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. 4

[22] 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. 9

[23] 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. 4

[24] 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

[25] 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. 4

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

[27] 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. 2, 3

[28] Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois C Knoll. Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception [best student paper award]. In *2023 IEEE Intelligent Transportation Systems ITSC*. IEEE, 2023. 11