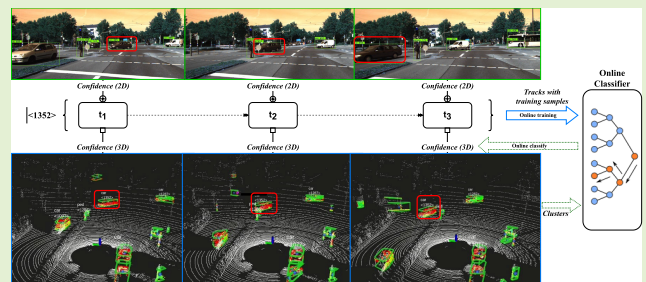


Efficient Online Transfer Learning for Road Participants Detection in Autonomous Driving

Rui Yang, *Student Member, IEEE*, Zhi Yan^{ID}, *Senior Member, IEEE*, Tao Yang^{ID}, *Member, IEEE*, Yaonan Wang^{ID}, and Yassine Ruichek^{ID}, *Senior Member, IEEE*

Abstract—The spatial information provided by 3-D object detection is critical for autonomous driving. Although visual-based object detection has made significant progress in 2-D/3-D in recent years, nonvisual sensors such as 3-D light detection and ranging (LiDAR) still have inherent advantages in the accuracy (ACC) of object localization. However, the challenge still lies in the interpretability of the sparse point clouds it generates and the difficulty of being manually annotated. In this article, we propose an online transfer learning framework based on multimodal sensor systems for 3-D object detection of urban road participants including pedestrians, cyclists, and cars. The framework aims to automatically and efficiently transfer object detection capabilities from 2-D monocular camera to 3-D LiDAR through a multitarget tracker-based pipeline, enabling knowledge transfer between sensors of different modalities. Furthermore, online random forest (ORF), an inherently fast and multiclass learning method, is innovatively integrated into our system. Experiments on two very different datasets, including KITTI and Waymo, demonstrate that the proposed framework not only rapidly builds 3-D detection capabilities about road participants in a single dataset but also maintains this capability across datasets. This reveals that the proposed framework is particularly suitable for in situ deployment of unmanned vehicles and can also solve the problem of insufficient generalization ability of current offline-trained detectors.

Index Terms—3-D object detection, autonomous driving, light detection and ranging (LiDAR), multimodal perception, online machine learning.



I. INTRODUCTION

OBJECT detection is an essential part of environment perception for autonomous driving, which can be further used for other tasks such as obstacle avoidance, trajectory

Manuscript received 10 July 2023; accepted 31 July 2023. Date of publication 18 August 2023; date of current version 2 October 2023. This work was supported in part by the China Scholarship Council through the CSC-UT-INSA Program, in part by the Contrat de Plan État-Région (CPER 2015–2020 Mobilitech), in part by the National Natural Science Foundation of China under Grant 62103332, and in part by the Jeunes Talents France-Chine Program. The associate editor coordinating the review of this article and approving it for publication was Dr. Brajesh K. Kaushik. (*Corresponding author: Zhi Yan.*)

Rui Yang, Zhi Yan, and Yassine Ruichek are with CIAD UMR 7533, Univ. Bourgogne Franche-Comté, UTBM, F-25200 Montbéliard, France (e-mail: rui.yang@utbm.fr; zhi.yan@utbm.fr; yassine.ruichek@utbm.fr).

Tao Yang is with the Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: yangtao@nwpu.edu.cn).

Yaonan Wang is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the National Engineering Research Center for Robot Visual Perception and Control Technology, Changsha 410082, China (e-mail: yaonan@hnu.edu.cn).

Data is available on-line at https://github.com/epan-utbm/efficient_online_learning.

Digital Object Identifier 10.1109/JSEN.2023.3305592

prediction, and social behavior analysis. In recent years, significant breakthroughs have been made in visual-based object detection methods, which can provide 2-D bounding boxes in images to determine the location and corresponding category of different objects [1], [2]. Although these 2-D information allow us to better understand the environment, it is far from sufficient for reaching autonomous driving in the real world. 3-D information such as the length, width, height, and deflection angle of objects is expected, which can provide more comprehensive environmental awareness for subsequent vehicle path planning and control strategies. The 3-D information can be estimated using multiple cameras [3], [4], [5]; however, the perception accuracy (ACC) based on this scheme is limited and inevitably affected by lighting conditions.

The application of 3-D light detection and rangings (LiDARs) in autonomous driving has been growing dramatically over the past ten years [6]. An important specification of this type of sensor is the ability to provide precise, long-range, and wide-angle distance measurements, and compared with visual sensors, LiDARs are robust to lightness variance. However, object detection in LiDAR scans can be very chal-

linging, because the representation of the sensory data to the entire environment is merely a set of sparse points without easy-to-learn color and texture features. More challenging, in response to the changes in the environment [7], [8] and different environments [9], [10], a feature-based classifier usually needs to be fine-tuned or even retrained to achieve acceptable performance in new scenarios. An approach to eliminate this dilemma is to train a generalized classifier. However, this requires a considerable amount of annotated data usually associated with high labor and machine costs. As for 3-D LiDAR data, labeling sparse points is tedious work and prone to human error, in particular if many variations in object pose, shape, and size need to be correctly classified.

Therefore, for both industry and academia, on one hand, it is necessary to save people from tedious data annotation and repetitive supervised training, and on the other hand, it is necessary for vehicles to have the ability to learn autonomously and continuously to adapt to most environmental changes, while human intervention is only required in rare edge cases. In this article, we propose a framework that can simultaneously satisfy the above two requirements by extending our proven online robot learning approach [11] to autonomous driving. The framework empowers the vehicle to efficiently and autonomously learn a high-performance 3-D LiDAR-based object detector in its deployed environment in a short period of time for 3-D detection of road participants including pedestrians, cyclists, and cars. In the field of autonomous driving, the environment in which vehicles operate is usually complex and changeable, which poses a huge challenge to the perception models based on offline learning (such as statistics). The online method updates the model by learning new samples in a timely manner and allows the model to be used while being trained, so as to adapt to the changes in the environment and different environments, thereby improving the long-term autonomy of the vehicle [12].

A conceptual diagram of our proposal is shown in Fig. 1. The core idea is to enable hard-to-learn point cloud-based detectors to benefit from off-the-shelf mature image-based detectors through an online knowledge transfer pipeline. Specifically, the point cloud generated by the 3-D LiDAR is first segmented into clusters, ideally each cluster corresponding to an object shown in the point cloud. Then, an online learning classifier (online random forest (ORF) [13]) makes probabilistic estimates of the categories to which these clusters belong. At the same time, an off-the-shelf detector is applied to the 2-D image generated by the camera to obtain the 2-D bounding boxes of the objects and their class confidence shown in the image. Afterward, the results from the detectors of different modalities are correlated using the defined calibration matrix between different sensors. Clusters with object class probabilities in different frames are associated by the multitarget tracker into their motion trajectories, and the probabilities of trajectories are computed to determine which object class they (and thus the corresponding clusters) belong to. Finally, these labeled

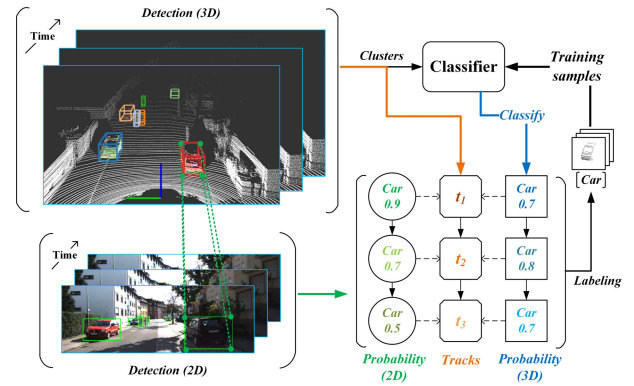


Fig. 1. Conceptual diagram illustrating the transfer learning process from images to point clouds. Initially, the detection outputs of the same object, including the detection of a pretrained visual-based 2-D detector (shown in green) and the detection of a nonvisual detector (shown in blue), are matched based on the spatial relationship of the objects in the scene. Subsequently, the detections in different point cloud frames are correlated using a multitarget tracker. The trajectories generated by the tracker (shown in orange) are probabilistically estimated to generate labeled samples required for online learning. Finally, a classifier is trained using ORF with the generated training samples, improving the object detection in 3-D space and forming a closed-loop system. This learning process is iterative.

clusters are input as training samples to the ORF for learning. The learning process iterates continuously with time and space changes and finally converges and remains stable.

This article is an extension of our previous work [14] and includes the following new contributions.

- 1) We provide a formal problem description on information fusion and further elaborate on the proposed online transfer learning framework, incorporating spatio-temporal association and closed-loop detection.
- 2) In the specific implementation, the output of the online learning detector becomes the input of the framework at the same time, truly realizing the “learning while using” as a closed-loop system.
- 3) We provide comprehensive experimental analyses, especially: 1) extending the evaluation of the object classification performance of the online learning detector in the previous paper [14] to the overall detection performance based on multisensor fusion detection and evaluating the collaborative detection performance of multimodal detectors; 2) evaluating our framework using the test set provided by KITTI and submitting the experimental results to the KITTI benchmark suite for community comparison; and 3) conducting cross-dataset (i.e., across-environment) experiments to evaluate the online adaptability of our proposed framework.

The remainder of this article is organized as follows. Section II gives an overview of related work, in particular on 3-D object detection and online machine learning. Section III formalizes the problem we studied. Then, we introduce our online learning framework in Section IV and the implementation details in Section V. Section VI provides a comprehensive experimental evaluation of the proposed framework. Finally, conclusions and future research are discussed in Section VII.

II. RELATED WORK

A. 3-D Object Detection

3-D object detection based on visual sensors has made great progress in recent years, thanks to the improvement of hardware computing capabilities and the development of software and algorithms, especially the great success of deep neural networks [3], [4], [5]. However, suffering from the limited ability of visual sensors to obtain reliable poses of objects and their sensitivity to lighting conditions, researchers have also been actively exploring detection methods based on other nonvisual sensors. For instance, Dequaire et al. [15] used a recurrent neural network (RNN) to capture the environment state evolution perceived by two 3-D LiDARs and trained a representation model in an unsupervised manner and used it to help self-driving cars detect and track the surrounding road participants. Chen et al. [16] proposed a method for segmenting moving objects in 3-D LiDAR point clouds and automatically generating their labels. The method uses a multitarget tracker to generate trajectories of moving objects to improve offline coarse detection results. Majer et al. [17] paid attention to the low performance of camera- and LiDAR-based detection in foggy conditions and proposed to use millimeter-wave radar and developed a pedestrian detector based on the support vector machine (SVM).

However, sensors of different modalities have their own strengths and weaknesses [7]. Relying solely on a single perception modality does not meet the needs of autonomous driving, especially in terms of safety. Therefore, multimodal sensor perception schemes are considered promising, and data fusion of different modalities has been extensively studied. For instance, González et al. [18] explored the fusion of RGB images generated by a monocular camera with point clouds generated by a 3-D LiDAR and trained a random forest (RF) for each view, and then combined all the RFs into a final ensemble for pedestrian, cyclist, and car detection. Based on the same sensor combination, Qi et al. [19] proposed to first generate 2-D object region proposals in RGB images, then extrude each 2-D region into a 3-D frustum to obtain the point cloud within the frustum from 3-D LiDAR data, and finally apply the PointNet [20] to estimate the 3-D bounding boxes from the points in the viewing frustum. This method is similar to ours in knowledge transfer, namely, from RGB camera to 3-D LiDAR. The main difference is that the former only uses instantaneous spatial information (i.e., detection), while we consider both spatial and temporal information (i.e., trajectories).

B. Online Machine Learning

The methods mentioned above are all offline learning methods except [17]. Offline learning requires the data to be complete and usually uses a considerable amount of manually annotated data to train a globally optimal model while allowing human intervention (such as fine-tuning training parameters and retraining the model). Moreover, the trained model does not change after being deployed on the agent. In contrast, online machine learning means that the agent can spontaneously and automatically update the knowledge

model over time, which does not contain assumptions about complete data, annotated data, human intervention, etc. [12]. Pioneering work in this area dates back more than two decades, and Thrun [21] advanced the idea that mobile robots should be capable of lifelong learning. In recent years, with the rapid development of various related technologies including hardware and algorithms, research on online learning for mobile robots has become more and more extensive [11], [17], [22], [23], [24], [25]. Among them, Teichman and Thrun [22] presented a semi-supervised learning method based on the expectation-maximization (EM) algorithm for the object trajectory classification problem in 3-D LiDAR data and showed that the learning of detection capabilities for dynamic objects can benefit from multitarget tracking systems. The proposed learning starts with a small set of hand-labeled seed trajectories and a large set of background trajectories precollected in areas free of any pedestrians, cyclists, or cars. In contrast, our research further focuses on in situ learning in the absence or with incomplete background knowledge, i.e., background samples are discriminated online as the model learns. In situ learning primarily focuses on training models directly on the target device or system, using local resources and data. The emphasis is on capturing the specific characteristics and context of the deployed environment. While online learning is a broader concept that encompasses the dynamic update of models as new data arrives, regardless of whether the training occurs on the target device or remotely.

On the other hand, ORF [13] is a method worth mentioning considering the simultaneous requirements of fast learning and multiclass classification for autonomous driving. It combines the idea of online bagging and extreme RF and proposes an online decision tree growth strategy, making the performance of online RF converge to that of offline RF.

C. Discussion

Through our investigation of the existing work, there is currently no effective way for a vehicle to efficiently and autonomously learn and update its detection model for road participants, especially dynamic objects. This need is particularly pronounced for 3-D LiDAR, which is commonly used in autonomous vehicles. Using transfer learning [26] can improve the learning process of a target predictive function by leveraging knowledge from a source domain and learning task for a target domain. The goal is to adapt the model to the target domain and task so that it generalizes well and performs better in the target setting. Transfer learning is often used offline when a sufficient amount of labeled data is available in both the source and target domains. In contrast, the online transfer learning approach is particularly useful when there is a lack of labeled data in the target domain or when the target domain is subject to changes over time. The choice between offline and online depends on the availability of labeled data, the nature of the target domain, and the specific requirements of the learning task.

An intuitive idea in our exploration of online transfer learning for mobile robots is to use visual sensors to train nonvisual sensors. The latter is learning in a self-supervised manner using unlabeled data with the assistance of annotations

from the former. The whole learning framework can be formalized as an online transfer learning paradigm. The reason for integrating ORF into our learning framework is that RF is inherently fast in training, suitable for multiclass learning, and easy to explain, which are in line with the expectations of the academic and industrial circles for autonomous vehicle components.

III. PROBLEM FORMULATION

A. Online Transfer Learning

According to [26], transfer learning can be formalized as follows. Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, and a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$. In the context of our research, a well-trained detector d_i detects an object x_i , denoted as $\mathcal{W} = (x_i, l_i, d_i)$, where $l_i \in L$ is the label provided by d_i , which forms a labeled set representing the results of learning task $\mathcal{T}_S$. In addition, the same object x_i could also be detected by another detector such as d_j using a different data source (i.e., target domain $\mathcal{D}_T$), but without a credible label (since d_j is not well-trained), resulting in $U = (x_i, l_j, d_j)$, where $l_j = \emptyset$. This set U is referred to as an unlabeled set, representing the promotable results of learning task $\mathcal{T}_T$, where $l_j \in L$ is a set of labels that need to be learned. The objective is to learn a detector d_j online $f: U \xrightarrow{W} L$ from the relationship between the labeled set $\mathcal{W}_x$ and the unlabeled set U_x .

B. Information Fusion

We further formalize $f: U \rightarrow L$ as an information fusion problem. Let Y denote the detections generated by a finite set of detectors $D = \{d_1, d_2, \dots, d_n\}$ that continuously process sensory data of various modalities such as point clouds, laser scans, and images over time. Let X be all the objects that can be detected by the detector in the environment, then each detection y can be defined as follows:

$$y_t^d = (x_i, l, b, d, t) \in Y \quad (1)$$

where x_i indicates the object detected at time t by detector d in which $x_i \in X_d$, and l and b denote the classification posterior about x_i given by detector d and its coordinates with respect to the sensor as the reference frame, respectively. The fusion of information extracted from different sensory data can thus be formalized as

$$\bigcap_{p=1, p \neq q}^n \{x_i\} = X_{d_p} \cap X_{d_q} \quad (2)$$

which represents an intersection of multiple detection of the same object x_i by different detectors.

IV. GENERAL FRAMEWORK

Following the formulation of the target problem, the proposed online transfer learning framework is shown in Fig. 2. It consists of four modules including an *off-the-shelf detector* (D_{off}), a *learn-by-use detector* (D_{learn}), a *multitarget tracker* (T_{multi}), and a *sample annotator* (A_{sample}). D_{off} refers to a

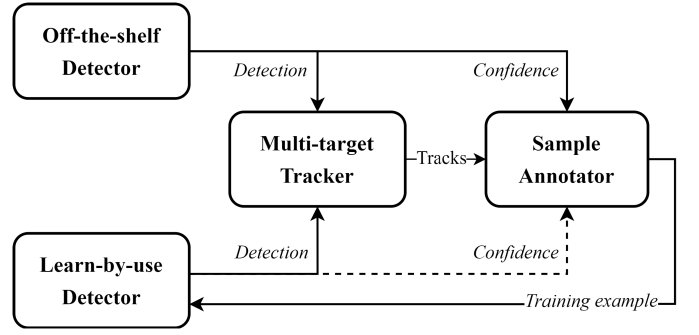


Fig. 2. Block diagram illustrates the overall framework of online transfer learning. The dotted line indicates that detection and confidence provided by learn-by-use detector will be further refined and learned online over time from the help of off-the-shelf detector.

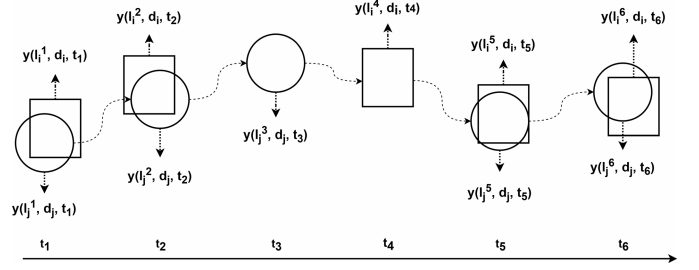


Fig. 3. Schematic of the working principle of the sample annotator. The squares represent one detector and the circles represent the other.

pretrained detector with high confidence. It can be further embodied as a visual sensor-based object detector in the context of this article. D_{learn} refers to the detector that needs to be learned online without human intervention. It is used as it learns and is continuously refined over time under the supervision of D_{off} . It can be embodied as a nonvisual sensor-based detector.

T_{multi} plays a key role in the framework, which spatiotemporally correlates detections from various detectors and actually builds a pipeline of knowledge output from D_{off} to D_{learn} . Its working principle can be formalized as

$$\text{Track}_{x_i} = \begin{bmatrix} y_{t-m}^{d_1} & \cdots & y_t^{d_1} & \cdots & y_{t+n}^{d_1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{t-m}^{d_i} & \cdots & y_t^{d_i} & \cdots & y_{t+n}^{d_i} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{t-m}^{d_k} & \cdots & y_t^{d_k} & \cdots & y_{t+n}^{d_k} \end{bmatrix}_{x_i} \quad (3)$$

The trajectory consists of the same object x_i at different times t_i^1 and there exist possible $k \in \mathbb{R}^+$ detectors that may be using different modalities of sensor data. Furthermore, the no-detection output is also a receivable message indicating that “detector d works at time t , but no object x_i is detected.”

A_{sample} estimates in real-time the class of each object track—in fact, all the object samples on this track—and then labels them and feeds them to D_{learn} for online learning. A visualization of the working principle of A_{sample} is shown in Fig. 3. Specifically, given an object track containing time-series detection samples from different detectors, the object

¹The m and n in the equation refer to different time instances.

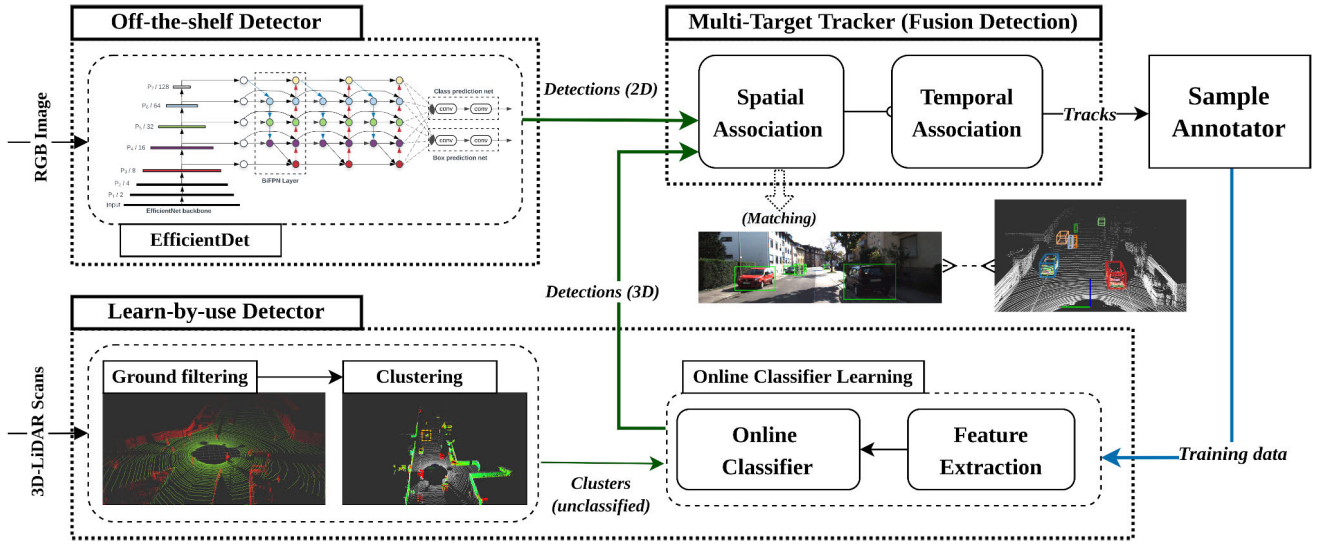


Fig. 4. Detailed system block diagram of efficient online transfer learning for object detection based on multimodal perception. The RGB image is fed to an off-the-shelf visual detector to get the bounding box of the object in the 2-D image and its probability of belonging to an object. At the same time, the point cloud generated by 3-D LiDAR is ground-filtered, segmented into different clusters and then sent to the multitarget tracker. In the latter spatial association, the 3-D bounding boxes of the clusters are projected into synchronized 2-D image frames to match the visual detection. The clusters are then input to a temporal association module to generate object trajectories. Next, the sample annotator estimates the probability that the trajectory belongs to a certain class, so as to determine the labels of all the samples on the trajectory, which is used for online learning of the learn-by-use detector. The green line represents detection, and the blue line represents learning, which together constitute the closed-loop structure of online learning.

can be detected by one or more detectors at any time t and the confidence of the category to which the object belongs is given at the same time. The labels of all the samples on the track are determined by fusing the confidences of all the samples. The samples provided by D_{learn} are sent back to itself for learning. The samples provided by D_{off} will be given final labels.

To be more specific, the learning process is initialized by A_{sample} , which will analyze the confidence of the objects on the whole track considering both $\mathcal{W}_w$ and U_u , and estimates the examples that are correctly or incorrectly classified to form the training set. Then, the training set is given to the classifier for training, i.e., the estimation of initial model θ_0 . After that, the learning process proceeds in an iterative bootstrap way. In iteration k , the classifier trained in the previous iteration classifies the entire unlabeled set

$$l_u^k = f(u|\theta_{k-1}), \quad u \in U_u. \quad (4)$$

These examples are added to A_{sample} and the labels are changed. At the end of the iteration, the classifier is retrained, i.e., θ_k is reestimated. This process is repeated until convergence or other stopping criteria. The key element of online transfer learning is A_{sample} 's estimation of object categories, and independent estimation of detection from different modality data allows their errors to be compensated. The performance of the learned classifier in the detector is assisted by auxiliary annotation provided from other detectors to learn from scratch in the processing of continuous data streams online. Meanwhile, the learned classifier is evaluated in each frame of the data stream, its errors are identified, and it is updated online to avoid the same errors in the future.

V. IMPLEMENTATION

In this section, we describe a concrete implementation based on the proposed general framework. On the one hand, this implementation is to solve the problem that 3-D LiDAR data are difficult to manually annotate in autonomous driving, and on the other hand, it enables 3-D LiDAR-based detectors to cope with environmental changes, across different environments or even long-tail problems. The implementation is fully based on ROS with a very high degree of modularity using C++ and Python mixed coding, which is publicly available to the community. The detailed module diagram of the proposed system is shown in Fig. 4, and each block will be introduced one by one below.

A. Off-the-Shelf Detector

The off-the-shelf image detector is based on the state-of-the-art deep learning methods to guarantee its detection performance. The detector detects cars, cyclists, and pedestrians in the 2-D image and outputs their 2-D bounding boxes and the corresponding detection confidences (i.e., categories and prediction scores). To strike a balance between the ACC and the real-time performance of the detector, we conducted comprehensive tests using three widely recognized methods including YOLOv3 [2], Faster R-CNN [1], and EfficientDet [27]. These methods were evaluated on the KITTI dataset [28]. The results are shown in Table I. The detailed configurations of training, validation, and test sets are given in Section VI.

Table I shows the time required for single-frame processing and the mean average precision (mAP) for three classes of object detection using different methods. It can be seen that YOLOv3 runs the fastest among the three, but the precision of detection is limited. Faster R-CNN achieves the best detection

TABLE I
PERFORMANCE EVALUATION OF THREE DEEP-LEARNING-BASED
2-D DETECTORS

Model	Car	Pedestrian	Cyclist	Runtime
YOLOv3 [2]	59.73%	38.51%	31.92%	21ms
Faster R-CNN [1]	81.57%	60.53%	57.83%	2635ms
EfficientDet (D1) [27]	75.83%	51.69%	46.92%	34ms

results, but at the expense of execution speed. EfficientDet strikes a good balance between the two metrics.

B. Learn-by-Use Detector

The detector that requires online learning adopts a pipeline structure instead of end-to-end because the former has better explainability and thus is more in line with the industrial needs of autonomous vehicles. In particular: 1) point clouds produced by 3-D LiDAR are first passed through a filter to remove the ground; 2) the filtered points are then segmented to generate distinct clusters; 3) next, feature extraction is performed on those clusters labeled as objects of interest; and 4) finally, the feature vector is sent to the online classifier for learning. Below we describe these four modules in detail.

1) *Ground Filtering*: Ground filtering can effectively reduce the burden of subsequent calculations and also improve the ACC of point cloud segmentation. Commonly used methods include thresholding [23] and fitting [29]. The former is simple to calculate and is suitable for flat ground or horizontally installed LiDAR. The latter is relatively complex in calculation but more robust to various terrains and sensor installation angles. The ground plane fitting (GPF) method proposed in [29] is adopted in our system. The method first divides the point cloud into segments evenly along the x -axis (vehicle driving direction). Then for each segment, a set of seed points with low height values are deterministically extracted to estimate the initial plane model of the segment's ground surface. This model is used to evaluate each point in the segment and generate the distance from that point to the orthographic projection on the candidate plane. Next, this distance is compared with a predefined threshold to determine whether the point belongs to the ground or not. Points belonging to the ground are used as seeds for a refined estimate of a new planar model, and the process is repeated a limited number of times. Finally, the entire ground is generated by concatenating the ground points extracted from each point cloud segment. Fig. 5 shows an example of removal results.

2) *Point Cloud Segmentation*: The clustering method proposed by Autoware [30] is used to segment the filtered point cloud. As input to this module, the nonground point cloud is defined as a set of I points

$$PC = \{pc_i | pc_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \dots, I\}. \quad (5)$$

To ease the computational burden of clustering, these points are first projected from 3-D to 2-D

$$PC' = \lambda PC, \quad \lambda = (1, 1, 0). \quad (6)$$

Clustering is then performed on PC' , and a cluster is defined as

$$C_j \subset PC', \quad j = 1, \dots, J \quad (7)$$

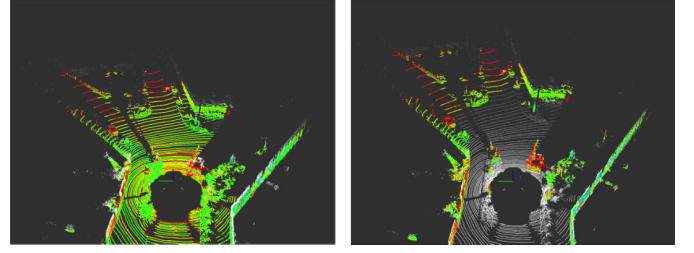


Fig. 5. Schematic of ground filtering. On the left is the original point cloud, and on the right is the result after using GPF. It can be seen that most of the points belonging to the ground are removed, and a small number of distant points representing slopes still exist.

TABLE II
FEATURES FOR CLASSIFIER TRAINING

Feature	Description	Dimension
f_1	Number of points included in the cluster	1
f_2	Minimum cluster distance from the sensor	1
f_3	3D covariance matrix of the cluster	6
f_4	Normalized moment of inertia tensor	6
f_5	Slice feature for the cluster	20
f_6	Reflection intensity distribution	27

where J is the total number of clusters, and

$$C_j \cap C_k = \emptyset, \quad \text{for } j \neq k, \text{ if } \min \|pc_j - pc_k\|_2 \geq \text{dis} \quad (8)$$

where the sets of points $pc_j, pc_k \in PC'$ belong to the point clusters C_j and C_k , respectively, and dis is a Euclidean distance threshold. Finally, a volume filter is used to filter out those clusters that are too large or too small

$$\bar{C} = \{C_j | L_{\min} \leq \text{len}_j \leq L_{\max} \\ W_{\min} \leq w_j \leq W_{\max}, H_{\min} \leq h_j \leq H_{\max}\} \quad (9)$$

where l_j , w_j , and h_j , respectively, represent the length, width, and height of the smallest cube² containing C_j . This filtering removes most background objects and thus improves the efficiency of subsequent modules in the pipeline.

3) *Feature Extraction*: Six features (a total of 61 dimensions, as shown in Table II) are extracted from the clusters for the next step of classifier training. The set of feature values of each sample C_j forms a vector $f_j = (f_1, \dots, f_6)$. Features from f_1 to f_4 were introduced by Navarro-Serment et al. [31], while features f_5 and f_6 were proposed by Kidono et al. [32]. The selection of these features is mainly based on the needs of real-time perception and lightweight on-board computing for autonomous driving [24]. For the performance comparison of each feature, refer to [24], [31], and [32].

4) *Online Classifier*: The ORF method [13] is applied for online classifier learning. Compared with the SVM used in our previous work [11], RF has the inherent advantages of fast training speed and support for multiclassification. The classic RF algorithm uses batch training to generate decision trees based on global data, which does not meet the needs of online learning. In contrast, ORF proposes an online decision tree growing procedure where when to split a node depends on: 1) whether there are enough samples in the node to have

²In our experiments, $L = [0.1, 5.5]$, $W = [0.1, 5.5]$, $H = [0.3, 5.5]$.

robust statistics and 2) whether the split is good enough for classification purposes. This idea is further concreted as

$$|R_j| > \alpha \wedge \exists s \in S : \Delta L(R_j, s) > \beta \quad (10)$$

where α is the minimum number of samples a node needs observe before splitting, β is the minimum gain the split needs to achieve, R_j is a decision node, and $\Delta L(R_j, s)$ is the gain of node j with respect to the test s , which is measured as

$$\Delta L(R_j, s) = L(R_j) - \frac{|R_{jls}|}{|R_j|} L(R_{jls}) - \frac{|R_{jrs}|}{|R_j|} L(R_{jrs}) \quad (11)$$

where R_{jls} and R_{jrs} are the left and right partitions based on s . Obviously, the idea is to find the test with the highest gain as a node splitting decision

$$s_j = \operatorname{argmax}_{s \in S} \Delta L(R_j, s). \quad (12)$$

Furthermore, on the basis of the original ORF, we added support for streaming data, mini-batch learning, and real-time model storage. For more details, refer to our released code.

C. Multitarget Tracker

1) *Spatial Association*: In our general framework, the multitarget tracker is designed to loosely connect detectors to achieve multisource information fusion. In the field of autonomous driving, the calibration between the various sensors and the hard synchronization of the data they generate are currently two of the de facto standard operations. Based on this fact and to maximize the performance of the implemented system, the 3-D detection bounding box in the point cloud is therefore first projected to the 2-D image and then compared with the native 2-D detection bounding box by measuring the intersection over union (IoU)³ to determine whether they are different detections of the same object. If so, the detections should be associated. This step can be regarded as the fusion of spatial information at time t . Followed is a pseudocode describing the spatial association algorithm.

Algorithm 1 Spatial Association at Time t

Input: Detected objects in image $\{O\}$, Clusters $\{\bar{C}\}$ and IoU threshold θ_{iou}

Output: Annotated clusters $\{C\}$

```

1: for each cluster  $\bar{C}_i$  in  $\bar{C}$  do
2:   Project  $\bar{C}_i$  to 2D space in image as  $Project_i$ 
3:   while there exists an object  $O_j \in \{O\}$  do
4:     if  $IOU(O_{bbox}, Project_i) \geq \theta_{iou}$  then
5:       let  $\theta_{iou} \leftarrow IOU(O_j^{bbox}, Project_i)$ .
6:       let  $\bar{C}_i^{label} \leftarrow O_j^{label}$ .
7:     end if
8:     Remove  $O_j$  from  $\{O\}$ .
9:   end while
10: end for
11: Return the clusters as  $\{C\}$ .
```

³In our experiments, we required an IoU of 0.7 for cars, and an IoU of 0.5 for pedestrians and cyclists, as per the KITTI benchmark suite.

2) *Temporal Association*: Second, detections at different times also need to be associated. This step can be done on 2-D images or 3-D point clouds. We choose the latter because 3-D LiDAR has a wider field of view and is robust to different lighting conditions, enabling us to associate more and better samples to ensure the efficiency of online learning. Specifically, the Autoware's multitarget tracker is integrated into our system for temporal association of samples, which is based on probabilistic data association (PDA), unscented Kalman filter (UKF), and interactive multiple models (IMMs). The PDA algorithm updates the target state according to the weighted sum of the probabilities of each possible target state update. The UKF method is a combination of the unscented transform (UT) and the standard Kalman filter. Through the UT transformation, the nonlinear system equations can be fit to the standard Kalman filter system under the assumption of linearity. This can effectively overcome the problems of low estimation ACC and poor stability, thus providing robust tracking performance.

The IMM allows constructing multiple motion models to perform the combined estimation for the motion states of the objects, which consists of four steps: interaction, filtering, model probability update, and combination of estimation. The state interaction embodies the essence of the method. Specifically, the input of a filter in model i is the combined estimation of each filter at the previous moment, and let the matching models at $k-1$ and k be j and i , respectively, then the equation of the combined probability is

$$\mu_{k-1|k-1}^{j|i} = \frac{1}{\hat{\mu}_{k|k-1}^i} \pi_{ji} \mu_{k-1}^j \quad (13)$$

where μ_{k-1}^j is the probability of model j at $k-1$; and π_{ji} denotes the probability that the object motion model transfers from model j to model i . The $\hat{\mu}_{k|k-1}^i$ is defined as

$$\hat{\mu}_{k|k-1}^i = \sum_{j=1}^n \pi_{ji} \mu_{k-1}^j. \quad (14)$$

Then the reinitialized states and covariances were estimated

$$\tilde{x}_{k-1|k-1}^i = \sum_{j=1}^n \hat{x}_{k-1|k-1}^j \mu_{k-1|k-1}^{j|i} \quad (15)$$

$$\tilde{P}_{k-1|k-1}^i = \sum_{j=1}^n \mu_{k-1|k-1}^{j|i} \left[\hat{P}_{k-1|k-1}^j + (\hat{x}_{k-1|k-1}^j - \tilde{x}_{k-1|k-1}^i)(-\tilde{x}_{k-1|k-1}^i) \cdot (\hat{x}_{k-1|k-1}^j - \tilde{x}_{k-1|k-1}^i)^T \right]. \quad (16)$$

Through the PDA-UKF-IMM combination, the trajectories of objects are identified by spatio-temporal association of detections from different detectors.

D. Sample Annotator

The sample annotator performs probabilistic fusion of samples associated with the multitarget tracker to determine in real-time which class of objects the samples on each track belong to. Specifically, the track probability [11] is applied, which is defined as follows. Let $P(l_i|x_i, d_j)$ denote the probability that example x_i is an object with its category label l_i

predicted by detector $d_j \in D$ at the precise time t , then the track probability $P(L_T|X_T, D)$ is computed by integrating the predictions of the different detectors according to the following formula:

$$P(L_T|X_T, D) = \frac{\text{odds}_{X_T}}{1 + \text{odds}_{X_T}} \quad (17)$$

where

$$\text{odds}_{X_T} = \prod_{i=1}^t \prod_{j=1}^K \text{odds}_{x_i}^j \quad (18)$$

and

$$\text{odds}_{x_i}^j = \frac{P(l_i|x_i, d_j)}{1 - P(l_i|x_i, d_j)}. \quad (19)$$

By thresholding⁴ $P(L_T|X_T, D)$, high-confidence samples are finally labeled (i.e., car, cyclist, or pedestrian) and fed to the learn-by-use detector for learning.

VI. EVALUATION

In this section, we evaluate the system implemented based on our proposed general framework from both macro and micro aspects. The former includes a set of end-to-end benchmarks (cf., Section VI-C) and experiments on online adaptability from the KITTI dataset to Waymo (cf., Section VI-F). The latter includes evaluation of the core module ORF (cf., Section VI-D) and ablation experiments for further analysis of the pipeline (cf., Section VI-E). All the experiments reported in this article were performed with Ubuntu 18.04 LTS (64-bit) and ROS Melodic, with an Intel i9-9900K CPU and 64-GB RAM, and a NVIDIA GeForce RTX 2080 GPU with 8-GB RAM.

A. Datasets

We conduct experimental evaluations of our online transfer learning framework on two open datasets including KITTI [28] and Waymo [33]. The KITTI dataset is a pioneering work on datasets in the field of autonomous driving. It mainly provides exteroceptive data in two modes of image and point cloud, and they are hard synchronized. Another important reason why KITTI has become a widely used dataset in the community is that it provides a benchmarking suite, which greatly simplifies the comparison between different methods. KITTI also has some limitations, as it looks at the problem of autonomous driving more from the perspective of computer vision than from the autonomous driving itself. For example, sensor data with different acquisition frequencies adopt one-to-one frame synchronization; labeling is completed in the image and then projected to the point cloud, which cannot truly reflect the geometric characteristics of the point cloud [34]; most scenes are conventional and low dynamic; weather and lighting conditions are good; and the object detection and tracking benchmark examples are discontinuous in time and do not contain global position information of the vehicle itself.

The Waymo dataset was first released in 2019 and highlights more autonomous driving features. It offers scene-based organization, providing continuous annotated data for each scene. Moreover, the dataset includes real-time global positioning information of the vehicle and uses many-to-one synchronization for data generated from different sensors. One notable aspect of the Waymo dataset is its diverse range of driving environments. It encompasses various weather conditions and different times of the day, allowing for a thorough evaluation of algorithms across different scenarios. An important motive for using this dataset is to assess the cross-environment learning capability of our proposed framework on the classification performance. Detailed experimental results are given in Section VI-F.

B. Experimental Settings

Since our framework deals with continuous and unannotated data, the raw data provided by KITTI are used for online transfer learning. Specifically, we select segments in the raw data that contain dynamic objects and exclude those that contain only monotonous scenes (where the objects are usually stationary vehicles and are mainly used for visual odometry tasks). Furthermore, since the sample distribution of road participants is unbalanced, i.e., there are many cars but few cyclists, an online downsampling of the learning samples of cars is therefore implemented to reduce the negative impact of data imbalance on learning. On the other hand, the 2-D images in KITTI's detection benchmark (consisting of three classes: car, cyclist, and pedestrian) are used to train the off-the-shelf camera-based detector and evaluate EfficientDet and other deep-learning-based methods via fivefold cross-validation (cf., Table I). For the latter, the provided 7481 nonsequential frames with ground-truth annotations are split in a ratio of 6:2:2 for training, validation, and testing, respectively. Whereas the 3-D point clouds in the detection benchmark are used to evaluate the performance of LiDAR-based detectors.

For the Waymo dataset, we randomly sample 15 clips from the training set by scene type for online updating of the classifier learned on KITTI. Since the format of the Waymo dataset is very different from that in KITTI, we first reorganized the original Waymo dataset into the KITTI format to facilitate data loading, model training, and evaluation.

C. ORF Versus RF

To evaluate the performance of the ORF module, we randomly selected 1500 samples from the training set (i.e., 60% of the 7481 point cloud frames) of KITTI to form a data stream as the input of ORF. Whenever ORF learns 100 samples, we save a model and evaluate it on the test set. We report results for the first 15 iterations. On the other hand, classical RF trained offline with the same 1500-sample batch was used as a baseline for comparison. The online and offline methods are consistent in the parameter setting of the forest structure, that is, trees = 100, depth = 50, epochs = 20, and split_threshold = 50. For other parameter settings, refer to the released code.

Average ACC and macro average (MaA) are used as performance evaluation metrics for classifiers. The former is

⁴The threshold was set to 0.7 in our experiments.

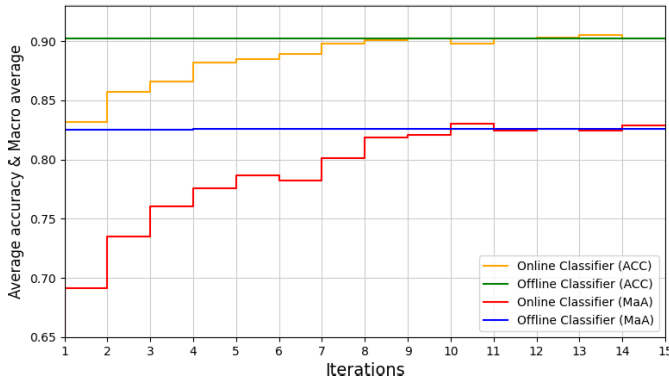


Fig. 6. Performance comparison of the ORF-based (online-trained) and the RF-based (offline-trained) 3-D LiDAR-based object classifiers.

calculated as follows:

$$ACC = 2 \cdot \frac{P \cdot R}{P + R} \quad (20)$$

while the latter is calculated as follows:

$$MaA = 2 \cdot \frac{P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}} \quad (21)$$

where for each category

$$P = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i}$$

$$R = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i} \quad (22)$$

and

$$P_{macro} = \frac{\sum_{i=1}^n P_i}{n}$$

$$R_{macro} = \frac{\sum_{i=1}^n R_i}{n} \quad (23)$$

where P is precision, R is recall, TP is true positive, FP is false positive, and FN is false negative.

The experimental results are shown in Fig. 6. It can be seen that as the number of training samples increases, the performance of online trained RF converges rapidly to that of offline trained and tends to be stable after the eighth iteration, in terms of both ACC and MaA. Moreover, compared with our previous work [14], the current implementation achieves closed-loop detection, where the detections generated by the point cloud detector are used to annotate the samples in the training phase. The resulting quality improvement of the learning samples ultimately improves the performance of the online learned detector.

D. Online Transfer Learning Performance

1) *Integrated-ORF Versus Standalone-ORF*: To evaluate the performance of our proposed system, we first compare the classification results of the 3-D LiDAR-based classifier (hereinafter Integrated-ORF) learned online with KITTI's raw data (without ground-truth annotations) under our framework (cf., Fig. 4), with the independently learned (with ground-truth annotations) ORF-based classifier (hereinafter Standalone-ORF) reported in Section VI-C. For the former, we still

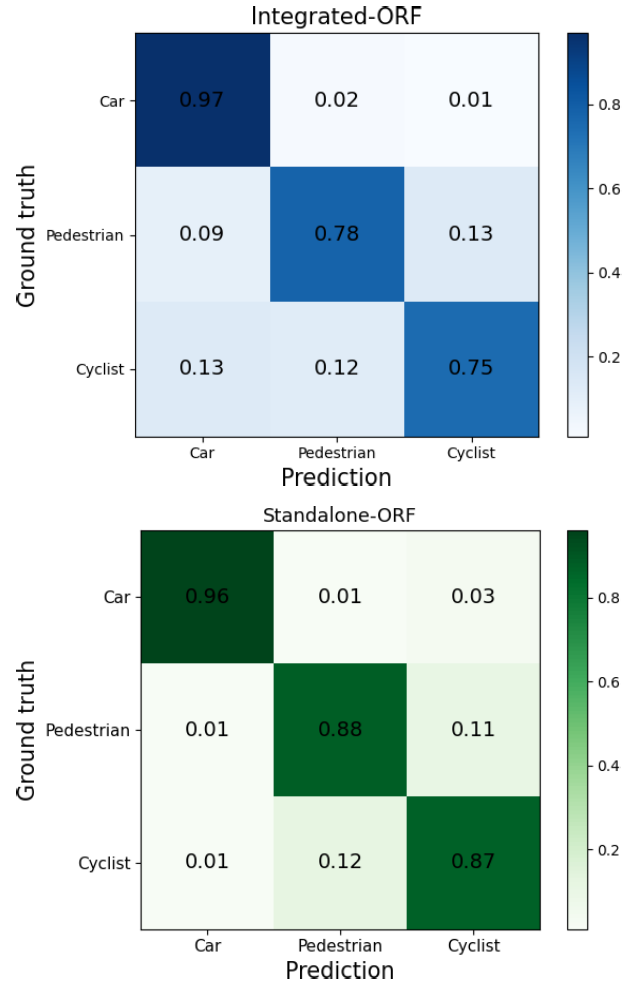


Fig. 7. Confusion matrix for performance comparison of Integrated-ORF and Standalone-ORF. The ordinate represents the true label, and the abscissa represents the predicted result. The darker the color, the higher the proportion of correct classifications.

train once when ORF obtains 100 samples and evaluate the model after the 15th iteration. Using the same test set as in the previous section, the comparison results are presented by confusion matrix as shown in Fig. 7.

It can be seen that the performance of Integrated-ORF does not catch up with Standalone-ORF in an all-round way, and the gap is concentrated in the classification of cyclists and pedestrians, which are the challenges in the perception of autonomous driving. In contrast, Integrated-ORF outperformed Standalone-ORF even slightly on car prediction. There are two reasons for this result. One is that the camera-based detector that plays the role of “trainer” has different detection capabilities for different object categories⁵ (cf., Table I). The second is that the point cloud clustering method based on unsupervised learning is currently not up to the ACC of manual segmentation, especially when it also includes projecting the point cloud in 3-D space to a 2-D plane (i.e., trading precision for speed), making it difficult for relatively small size objects to be accurately segmented [24]. These two reasons cause

⁵Note that we use a multiclass detector instead of some of the top-ranked single-class detectors on KITTI.

TABLE III
EVALUATION RESULTS USING AP ON THE KITTI 3-D OBJECT DETECTION VALIDATION SET

Method	Cars(%)			Pedestrians(%)			Cyclists(%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Learn-by-use detector	65.12	58.89	46.52	38.67	34.37	23.34	50.73	34.14	26.67
+ Off-the-shelf detector	80.45	67.73	58.81	47.67	40.32	35.44	75.61	56.55	47.61

TABLE IV
EVALUATION RESULTS USING AP ON THE KITTI 2-D OBJECT DETECTION VALIDATION SET

Method	Cars(%)			Pedestrians(%)			Cyclists(%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Off-the-shelf detector	82.16	78.18	67.09	60.25	49.97	44.84	55.18	46.23	39.35
+ Learn-by-use detector	93.87	90.45	78.47	82.59	70.13	61.25	80.25	75.31	62.87

TABLE V
EVALUATION RESULTS USING AP ON THE KITTI 3-D OBJECT DETECTION TEST SET (FOR RANKING)

Method	Cars(%)			Pedestrians(%)			Cyclists(%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [35]	74.97	63.63	54.00	-	-	-	-	-	-
AVOD [36]	76.39	66.47	60.23	36.10	27.86	25.76	57.19	42.08	38.29
F-Pointnet [19]	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
AVOD-FRN [36]	82.11	71.70	67.08	50.46	42.27	39.04	63.76	50.55	44.93
PFF3D [37]	81.11	72.93	67.24	43.93	36.07	32.86	63.27	46.78	41.37
Ours	80.08	67.67	58.68	48.22	39.96	34.67	73.72	56.64	47.94
+ Size estimation	79.97	69.13	58.57	48.65	40.11	35.99	75.20	58.96	50.41

TABLE VI
EVALUATION RESULTS USING AP ON THE KITTI 2-D OBJECT DETECTION TEST SET (FOR RANKING)

Method	Cars(%)			Pedestrians(%)			Cyclists(%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [35]	93.08	84.39	79.27	-	-	-	-	-	-
AVOD [36]	95.17	89.88	82.83	50.90	39.43	35.75	66.45	52.60	46.39
F-Pointnet [19]	95.85	95.17	85.43	89.83	80.13	75.05	86.86	73.16	65.21
AVOD-FRN [36]	94.70	88.92	84.13	67.95	57.87	55.23	70.38	60.79	55.37
PFF3D [37]	95.37	92.15	87.54	62.12	52.53	50.27	79.44	66.25	60.11
Ours	93.11	91.20	78.22	83.08	69.97	60.98	81.99	74.90	63.06
+ Size estimation	96.31	91.17	81.20	84.74	71.45	64.58	85.62	76.88	66.04

Integrated-ORF to learn some false positive samples. However, it is still worth pointing out that Integrated-ORF is learned without ground truth and human intervention.

2) *From Classification to Detection and Single Modality Versus Multimodality*: Object detection is a joint result of both classification and localization. To evaluate the performance of the learn-by-use detector, we conducted experiments using the test set (i.e., 20% of the 7481 point cloud frames), and the results are shown in the first row of Table III. The evaluation metric used is the unified standard provided by KITTI, namely, the average precision (AP). The idea is to compress the information in the precision-recall curve into a single number, thus making the comparison of different methods intuitive. The AP score is calculated by summing the precision values of 11 equally spaced recall values. For each object category, KITTI defines three detection difficulties including “easy,” “moderate,” and “hard,” according to the height of the bounding box, occlusion, and truncation. For example, in the “easy” level, objects have larger bounding boxes with less occlusion and truncation.

On the other hand, to verify the hypothesis that multimodal detection outperforms unimodal detection, we further evaluate the performance of the fusion of the learn-by-use

detector and the off-the-shelf detector, and the results are shown in Table III. Specifically, they are the output of the “Sample Annotator” that has not undergone “Temporal Association,” since the test set consists of nonconsecutive frames. Furthermore, the additional experiments illustrate that once the learn-by-use detector has competitive detection ability, it can even help the off-the-shelf detector in turn to improve the detection results in 2-D images, which is supported by Table IV.

3) *Comparison With Other Methods*: The test set (without ground truth) provided by KITTI for public ranking is used to compare our results with others in the community. Five representative methods are selected from KITTI’s ranking table for comparison. The principles of selection are: 1) methods using both image and point cloud data; 2) highly cited methods (MV3D [35], F-Pointnet [19], AVOD and AVOD-FRN [36]); and 3) recent methods (PFF3D [37]). The experimental results are shown in Tables V and VI, respectively, where ours is the multimodal detector mentioned above.

As can be seen from Table V, our online learning framework exhibits competitive performance compared with other offline methods, especially in the category “cyclists” we achieved the best results in easy and moderate difficulty.

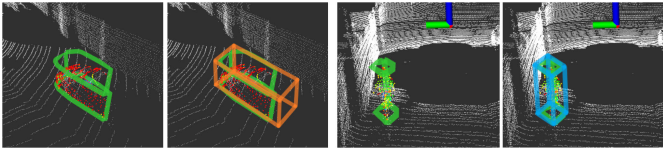


Fig. 8. Schematic illustration of 3-D object size estimation in point clouds. Through size estimation, the original convex box is inflated to the bounding box. The left two images show a car, while the right two images show a pedestrian.

On the other hand, although our method fails to achieve the best results on the two categories of “car” and “pedestrian,” the gap is not significant. Taking moderate difficulty as an example, ours is 5.26% and 2.31% away from the best performers, respectively. The reason, on one hand, is that the clustering method we use suffers from the accurate segmentation of road participants, which leads to some missed detections and false alarms. On the other hand, if some clusters are too small in the 3-D point cloud, they will be ignored due to their low IoU value after being projected into the 2-D image.

Correspondingly, Table VI shows the object detection performance of our multimodal detector on 2-D images in comparison to other methods. It can be seen that for moderate difficulty, ours still wins on the cyclist detection and is only 3.97% away from the best performance on car detection. As for pedestrian, although we are 10.16% behind the first place, we are still in the top 7% of the entire KITTI ranking.

4) *Size Estimation*: KITTI’s object annotations in point clouds are based on projections from those in 2-D images, thus including an estimate of the size of the object in its 3-D bounding box. In contrast, point cloud clustering methods commonly used in autonomous driving, such as the Autoware method we use, output the true size of the clusters. Based on this fact and to present a fairer comparison with other methods, we additionally give the results of our multimodal detector after adding size estimation in Tables V and VI. Specifically, we randomly sample the training set to estimate volumetric models for each object class, and if a cluster is classified, then the convex box representing its size is inflated to the bounding box defined by its volume model. An intuitive result is shown in Fig. 8.

As can be seen from Tables V and VI, after adding size estimation, the detection performance of each category of objects shows an overall improvement, which is especially obvious in the detection of cyclists. This is mainly due to the special structure of cyclists; they usually contain a vertical person and a horizontal bicycle, so their bounding boxes defined in the point cloud by polygons are usually smaller than the manual annotations provided by KITTI. Furthermore, it is worth pointing out that if the test set consists of consecutive frames, then the “Temporal Association” module in our multitarget tracker can be used to estimate the true size of objects, thus sidestepping the need for empirical volumetric models.

E. Ablation Study

The ablation experiment aims to study the contribution of each module in a system to the overall performance of the

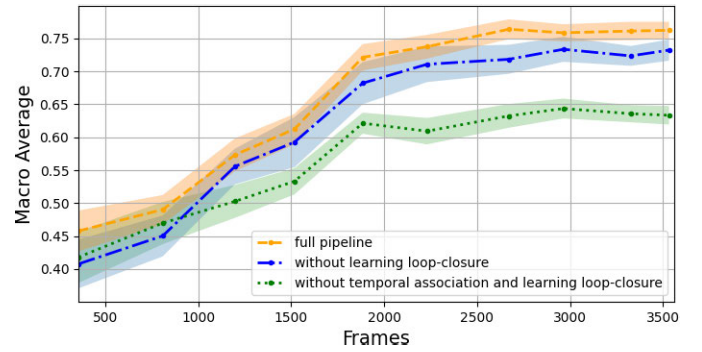


Fig. 9. Ablation experimental results.

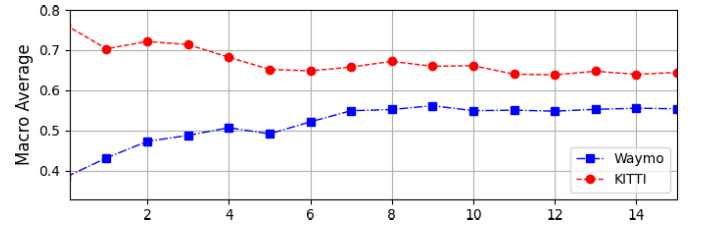


Fig. 10. Changes in classification performance after deploying the 3-D object detector learned on the KITTI dataset to the Waymo dataset and continuing to learn. The blue line represents the results evaluated on the Waymo test set. The red line indicates the results evaluated on the KITTI test set.

system. In our case, the classification performance of detectors based on different system architectures is evaluated after each iteration in the online learning process using the MaA metric. The experimental results are shown in Fig. 9. It can be seen that simply introducing 2-D image detection to match point cloud clusters in 3-D space can make the online learning performance present an overall upward trend (shown by the green line), but the improvement is not obvious after the environment becomes complex (starting from about 1000 frames, more pedestrians and cyclist appear). This situation is improved by introducing temporal association to the samples (shown by the blue line).

In addition, the classification performance of the detector is also improved overall. This is mainly because the temporal association makes those clusters that are not matched by the spatial association likely to become learning samples later, thus improving the learning efficiency. In addition, by introducing the learning loop closure, i.e., the probability estimation of the learn-by-use detector to the class of the point cloud cluster is also input into the multitarget tracker and then becomes a part of the learning, so that the online learning performance is improved again (shown in orange line).

F. Learn on KITTI Test on Waymo

Learning across environments is an important capability of our proposed learning framework. To verify this claim, we deploy the learn-by-use detector to the Waymo dataset and update it online. We selected 15 clips (approximately 2970 images and LiDAR scans) with the largest difference from KITTI in the Waymo dataset for learning. The MaA metrics are used, and the experimental results are shown in Fig. 10. It can be seen that due to the significantly different

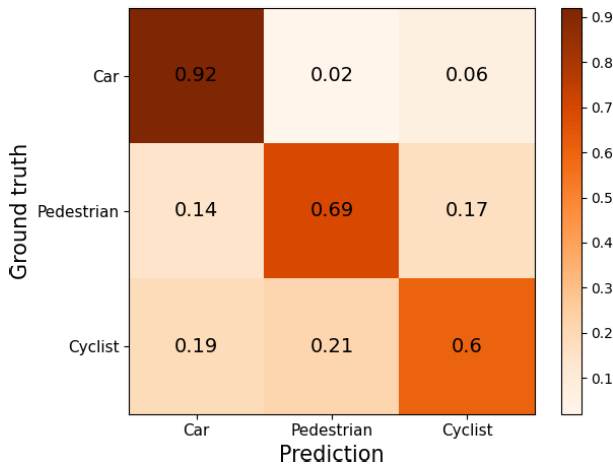


Fig. 11. Confusion matrix representing the 3-D object detector performance after learning on the Waymo dataset. The ordinate represents the true label, and the abscissa represents the predicted result. The darker the color, the higher the proportion of correct classifications.

driving scenarios exhibited by the KITTI and Waymo datasets, the classification performance of the detector shows a cliff-like decline at the beginning. However, thanks to our framework, with the input of Waymo data, the classification performance of the detector improves steadily with the learning iterations and finally stabilizes after the eighth iteration. On the other hand, evaluating the updated model with the Waymo dataset back to the test set of KITTI found that its performance overall decreased with learning iterations. This is a manifestation of catastrophic forgetting, which will be the focus of our next research.

The final classification performance of the detector after 15th iterations is shown in Fig. 11. It can be seen that the overall performance has declined compared with the one reported in Section VI-D1, where the drops are 0.05, 0.09, and 0.15 for cars, cyclists, and pedestrians, respectively. This is primarily due to the fact that if the scenes selected from Waymo are simple, the detector trained on the KITTI dataset performs admirably, but it fails to effectively demonstrate the strengths of our proposed online learning framework. Therefore, the selected clips containing complex scenes pose a huge challenge to the segmentation of the point cloud and the object tracking algorithm in it, which directly affects the quality of the learning samples. It is worth pointing out that the main purpose of our experiments on the Waymo dataset is to demonstrate the online adaptability of our framework across environments.

G. Qualitative Analysis

It can be seen from Fig. 12 that almost all the road participants are correctly detected in the matched image and point cloud and tracked in the latter. The section highlighted by the blue box in the point cloud demonstrates that even though the vehicle is no longer within the camera's field of view, it is still a learnable sample being tracked. And the range box area indicates that despite being detected as a cyclist in the image, the object is correctly learned as a pedestrian by the classifier through spatio-temporal association. While one exception is

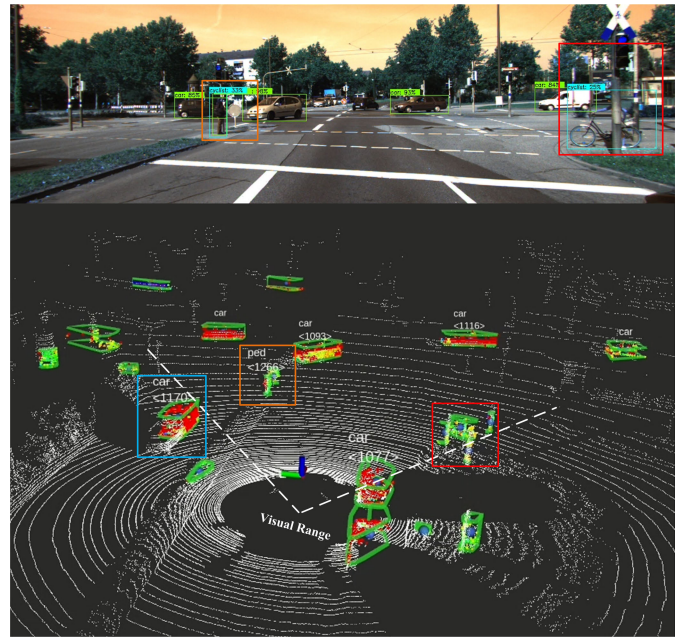


Fig. 12. Synchronized pair of images and point clouds from the KITTI dataset. The labels in the images contain the class information, while the labels in the point clouds contain both the category and Track-ID. The colored points enclosed by labels and bounding boxes represent the samples used for online learning.

the part annotated by the red box, where the cyclist and the utility pole are too close together, leading to their aggregation as a single entity (i.e., under-segmented), resulting in the failure of correct detection. This is a challenge both visually and in 3-D LiDAR. One possible solution to improve this issue is to incorporate color and texture information into the point cloud using pixel-to-point matching for further segmentation.

VII. CONCLUSION AND FUTURE WORK

In this article, we presented an efficient online transfer learning method for 3-D detection of road participants in autonomous driving. The core idea of this method is to use a 2-D visual sensor (i.e., a monocular camera) to help on-the-fly a 3-D nonvisual sensor (i.e., a 3-D LiDAR) learn multiclass object detection skills. Specifically, using the proven multitarget tracker provided by Autoware, a pipeline is built between the off-the-shelf detector and the learn-by-use detector to achieve knowledge transfer, and the ORF algorithm is innovatively integrated to achieve multiclass fast incremental learning.

Through comprehensive experiments, it is proven that our proposed framework can not only enable 3-D LiDAR to quickly learn the detection ability of road participants (i.e., cars, pedestrians, and cyclists) in the urban environment without ground truth and human intervention but also leverage multimodal detectors to improve the overall detection of road participants in autonomous vehicles. The ranking achieved by submitting the experimental results to the KITTI benchmark shows that the proposed online learning system is competitive in both 2-D and 3-D object detection performance, especially in the categories of pedestrians and cyclists, surpassing most existing methods. In addition, experiments across datasets also

demonstrate the adaptability of our learning framework to different environments, making it particularly suitable for in situ deployment. Certainly, continuous learning always requires combating knowledge forgetting. Our experimental results also reveal this challenging problem, which will be the focus of our next work. Furthermore, using the proposed framework to improve the tracking of dynamic road participants is also one of our future works.

REFERENCES

- [1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1137–1149, May 2017, doi: 10.1109/tpami.2016.2577031.
- [2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [3] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.
- [4] A. Mousavian, D. Anguelov, J. Flynn, and J. Koščeká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5632–5640.
- [5] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2345–2353.
- [6] R. Qian, X. Lai, and X. Li, "3D object detection for autonomous driving: A survey," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108796.
- [7] Z. Yan, L. Sun, T. Krajník, and Y. Ruichek, "EU long-term dataset with multiple sensors for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10697–10704.
- [8] T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, "FreMEN: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 964–977, Aug. 2017.
- [9] Z. Yan, S. Schreiberhuber, G. Halmetschlager, T. Duckett, M. Vincze, and N. Bellotto, "Robot perception of static and dynamic objects with an autonomous floor scrubber," *Intell. Service Robot.*, vol. 13, no. 3, pp. 403–417, Jul. 2020.
- [10] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5942–5948.
- [11] Z. Yan, L. Sun, T. Duckett, and N. Bellotto, "Multisensor online transfer learning for 3D LiDAR-based human detection with a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7635–7640.
- [12] Z. Yan, L. Sun, T. Krajník, T. Duckett, and N. Bellotto, "Towards long-term autonomy: A perspective from robot learning," 2022, *arXiv:2212.12798*.
- [13] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 1393–1400.
- [14] R. Yang, Z. Yan, Y. Ruichek, and T. Yang, "Efficient online transfer learning for 3D object classification in autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2950–2957.
- [15] J. Dequaire, P. Ondruška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 492–512, Apr. 2018.
- [16] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6107–6114, Jul. 2022.
- [17] F. Majer, Z. Yan, G. Broughton, Y. Ruichek, and T. Krajník, "Learning to see through haze: Radar-based human detection for adverse weather conditions," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2019, pp. 1–7.
- [18] A. González, D. Vázquez, A. M. López, and J. Amores, "On-board object detection: Multicue, multimodal, and multiview random forest of local experts," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3980–3990, Nov. 2017.
- [19] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.
- [20] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
- [21] S. Thrun, "A lifelong learning perspective for mobile robot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jan. 1994, pp. 23–30.
- [22] A. Teichman and S. Thrun, "Tracking-based semi-supervised learning," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 804–818, 2012.
- [23] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3D LiDAR-based tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 864–871.
- [24] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for 3D LiDAR-based human detection: Experimental analysis of point cloud clustering and classification methods," *Auto. Robots*, vol. 44, no. 2, pp. 147–164, Jan. 2020.
- [25] G. Broughton, F. Majer, T. Rouček, Y. Ruichek, Z. Yan, and T. Krajník, "Learning to see through the haze: Multi-sensor learning-fusion system for vulnerable traffic participant detection in fog," *Robot. Auto. Syst.*, vol. 136, Feb. 2021, Art. no. 103687.
- [26] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [27] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.
- [28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [29] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5067–5073.
- [30] S. Kato et al., "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPs)*, Apr. 2018, pp. 287–296.
- [31] L. E. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional lidar data," in *Proc. 7th Conf. Field Service Robot. (FSR)*, 2009, pp. 103–112.
- [32] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian recognition using high-definition LiDAR," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 405–410.
- [33] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.
- [34] T. Yang et al., "3D ToF LiDAR in mobile robotics: A review," 2022, *arXiv:2202.11025*.
- [35] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6526–6534.
- [36] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.
- [37] L.-H. Wen and K.-H. Jo, "Fast and accurate 3D object detection for LiDAR-camera-based autonomous vehicles using one shared voxel-based backbone," *IEEE Access*, vol. 9, pp. 22080–22089, 2021.



Rui Yang (Student Member, IEEE) received the M.S. degree in computer science and technology from Beijing Technology and Business University (BTBU), Beijing, China, in 2019. He is pursuing the Ph.D. degree with the Distributed Artificial Intelligence and Knowledge Laboratory (CIAD), University of Technology of Belfort-Montbéliard (UTBM), Belfort, France. His current research interests include robot perception and learning for autonomous driving.



Zhi Yan (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Paris 8 University, Paris, France, in 2006, 2008, and 2012, respectively.

From 2013 to 2015, he was a Postdoctoral Research Fellow in Software Engineering for Mobile Robotics with the CAR Team, École des Mines de Douai, Douai, France. From 2016 to 2017, he was a Postdoctoral Research Fellow with the Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln, Lincoln, U.K.,

mainly working on the Horizon 2020 Project FLOBOT. Since 2017, he has been an Assistant Professor (*Maitre de Conférences*) with the University of Technology of Belfort-Montbéliard (UTBM), Belfort, France. His current research interests include robot perception and learning for human-aware navigation and long-term autonomy.



Yaonan Wang received the B.S. degree in computer engineering from the East China University of Science and Technology, Fuzhou, China, in 1981, and the M.S. and Ph.D. degrees in control engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively.

He was a Postdoctoral Research Fellow with the National University of Defense Technology, Changsha, from 1994 to 1995; a Senior Humboldt Fellow in Germany, from 1998 to 2000; and a Visiting Professor with the University of Bremen, Bremen, Germany, from 2001 to 2004. He has been a Professor with Hunan University since 1995. His research interests include robot control, intelligent control and information processing, industrial process control, and image processing.

Prof. Wang has been an Academician of the China Engineering Academy since 2019.



Tao Yang (Member, IEEE) received the B.S. degree in detection, guidance, and control technology and the M.S. degree in control engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 2011 and 2014, respectively, and the Ph.D. degree in automation from the University of Technology of Belfort-Montbéliard (UTBM), Belfort, France, in 2019.

From 2019 to 2020, he was a Postdoctoral Research Fellow with the Distributed Artificial Intelligence and Knowledge Laboratory (CIAD), UTBM. Since 2020, he has been an Associate Professor with the Unmanned System Research Institute (USRI), NPU. His research interests include computer vision, point cloud processing, and chronorobotics.



Yassine Ruichek (Senior Member, IEEE) received the Ph.D. degree in control and computer engineering and the Habilitation à Diriger des Recherches degree in physical science from the University of Lille, Lille, France, in 1997 and 2005, respectively.

Since 2007, he has been a Full Professor with the University of Technology of Belfort-Montbéliard (UTBM), Belfort, France. His research interests are concerned with multi-sensory data-based perception and localization,

including computer vision, pattern recognition and classification, machine learning, and data fusion, with applications to intelligent transportation systems and video surveillance.