HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR

Zonglin Meng, Xin Xia, Runsheng Xu, Wei Liu, and Jiaqi Ma, *Member, IEEE*

***Abstract*—3D-LiDAR-based cooperative perception has been generating significant interest for its ability to tackle challenges such as occlusion, sparse point clouds, and out-of-range issues that can be problematic for single-vehicle perception. Despite its effectiveness in overcoming various challenges, cooperative perception's performance can still be affected by the aforementioned issues when Connected Automated Vehicles (CAVs) operate at the edges of their sensing range. Our proposed approach called HYDRO-3D aims to improve object detection performance by explicitly incorporating historical object tracking information. Specifically, HYDRO-3D combines object detection features from a state-of-the-art object detection algorithm (V2X-ViT) with historical information from the object tracking algorithm to infer objects. Afterward, a novel spatial-temporal 3D neural network performing global and local manipulations of object-tracking historical data is applied to generate the feature map to enhance object detection. The proposed HYDRO-3D method is comprehensively evaluated on the state-of-the-art V2XSet. The qualitative and quantitative experiment results demonstrate that the HYDRO-3D can effectively utilize the object tracking information and achieve robust object detection performance. It outperforms the SOTA V2X-ViT by 3.7% in AP@0.7 of object detection for CAVs and can also be generalized to single-vehicle object detection with 4.5% improvement in AP@0.7.**

***Index Terms*—Cooperative driving automation, cooperative perception, object detection and tracking, LiDAR.**

I. Introduction

COOPERATIVE driving automation (CDA), as standardized by SAE J3216 [1], aims at combining V2X communication and automated vehicles to enable real-time cooperation between connected vehicles, road users, and infrastructure to improve the safety, mobility, environmental sustainability, situational awareness, and operational efficiency of traffic flow [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Cooperative object detection and tracking in cooperative perception is one of the most critical modules in the CDA platform [14], [15], [16], [17], [18], [19], [20]. Similar to the single vehicles perception [21], [22], the task of cooperative object detection and tracking is to detect surrounding objects by sharing the sensor information and provide the objects' identity (ID), speed, and orientation information for the ego vehicle to facilitate other modules, including prediction, planning, and control [23], [24], [25], [26], [27], [28], [29], [30]. 3D-LiDAR is a commonly used advanced sensor in CAVs to perceive their surrounding environment. Given its accurate 3D information, object detection and tracking using LiDAR point clouds have attracted substantial attention with the rapid development of automated driving and CDA technologies [31], [32], [33], [34], [35]. However, the occlusion, sparsity of point cloud, and out-of-range issues are still constraining the performance of LiDAR-based object detection and tracking, which are the main challenges to reliable cooperative perception systems [36].

Based on 3D-LiDAR, the object detection and tracking framework can be classified into two categories: tracking-by-detection-based and joint-detection-tracking-based. In the object tracking-by-detection-based approach, the objects in each LiDAR frame are first detected as bounding boxes and then given the detection results, the object tracking associates the objects and estimates the trajectories of each object [37], [38] across frames. The object detection results are generally obtained from a deep neural network [39], [40], [41]. The widely used two-stage detector PointRCNN [42], which takes the single point cloud frame, generates the 3D bounding boxes proposal from segmented foreground points at the first stage and then refines the proposals at the second stage. SECOND [43] follows the VoxelNET [44] method to generate the point cloud feature for each voxel and then utilizes the sparse convolution to speed up the inference. Then, the object association in the tracking algorithm is usually formulated as a bipartite matching problem to track the detected objects. With the application of 3D LiDARs, [45] propose a 3D intersection over union (IoU) metric for object association. [46] calculates the association between detection results and trajectories by using the Mahalanobis distance. [47] matches the detection results with high-confidence score trajectories and then matches the remaining detection results with low-confidence score trajectories. [48] measures the association by considering both the motion model and the appearance cost. The results in [49], [50] show the state-of-the-art performance under challenge scenarios [51], [52].

Although the tracking-by-detection framework has shown good performance when the object detection module can detect

Manuscript received 16 May 2023; accepted 26 May 2023. Date of publication 12 June 2023; date of current version 22 September 2023. This work was supported in part by RIMI: Infrastructure-assisted Automation by California Statewide Transportation Research Program (SB 1) Program and in part by the Cooperative Perception and Control for Freeway Traffic System Operations by Federal Highway Administration Exploratory Advanced Research Program. *(Corresponding author: Xin Xia.)*

Zonglin Meng, Xin Xia, Runsheng Xu, and Jiaqi Ma are with the Department of Civil and Environmental Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: meng925@g.ucla.edu; x35xia@ucla.edu; rxx3386@g.ucla.edu; jiaqima@ucla.edu).

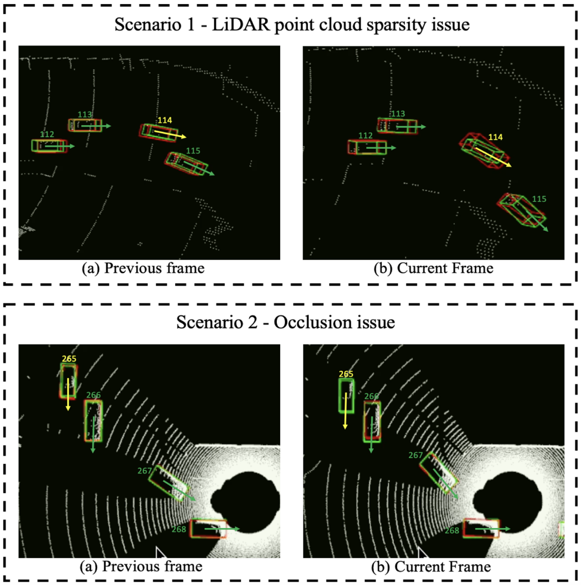
Wei Liu is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: liuw1619@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3282567.

Digital Object Identifier 10.1109/TIV.2023.3282567

objects normally, this framework treats object detection and tracking as independent tasks and the performance of object detection purely relies on the individual LiDAR frame because no historical information from past LiDAR frames is considered. Then, errors from falsely-detected or miss-detected objects due to the occlusion, sparsity of point cloud, and out-of-range issues will be propagated to the object tracking algorithm inevitably. One of the approaches to address the occlusion, sparsity of point cloud, and out-of-range issues is to leverage the shared information from other CAVs in a cooperative perception framework [14]. This kind of method just extends the sensing range through the shared information from other vehicles. However, at the boundary of the sensing range, the cooperative perception algorithm still suffers from the out-of-range issue and the detected objects are not stable. Another feasible way is to exploit the historical information from the object tracking algorithm as the correctly tracked objects' information contains the geometry information such as size or shape, and kinematics, i.e. trajectories, of the corresponding objects. Intuitively, these pieces of prior information from object tracking will benefit object detection even for occluded or faraway objects as the detector may acquire clues from their historical trajectories to make proper inferences.

Compared to the tracking-by-detection framework, the joint object detection and tracking framework attempts to use short-term historical information to address the drawbacks without considering the historical information in the former framework to some extent. The joint object detection and tracking framework regards the object detection and tracking problem as an end-to-end task by using deep neural network (DNN), which generally takes the current LiDAR frame along with its adjacent previous frame of the point cloud as inputs to a DNN to detect and track the objects simultaneously [53], [54], [55], [56]. In other words, this framework replaces the kinetic model with a branch of the neural network to predict the movement of the vehicles [53], [54]. However, this framework usually just takes into account short-term historical information from only one adjacent previous frame and is still sensitive to the occlusion, sparsity of point cloud, and out-of-range issues, such as the failure of association due to miss detection in the prior or current frame. Fig. 1 shows the common problems of current object detection issues. When a tracked object moves away from the ego vehicle, the joint object detection and tracking algorithm can generally track the vehicle when it is close to the ego vehicle but as the vehicle drives further away, the difficulty and uncertainty of the object detection increase, which likely results in problems such as the varying size of bounding boxes, miss detection, and erroneous detection. In [57], a comprehensive survey conducted revealed that infrastructure-based sensors can further enhance the object detection and tracking performance and in [58], a heterogeneous cooperative perception method using LiDAR point cloud data is proposed to fuse the deep feature for object detection efficiently. Involving the infrastructure-based sensors to some extent improve the object detection and tracking performance in that challenging scenario but still can not resolve it. In other words, joint object detection and tracking only considers the consecutive two frames, and failure from any frame of these two will lead to a fault in object detection and tracking especially when the point cloud of the surrounding vehicle is sparse. The limitations of the existing joint object detection and tracking algorithm motivate us to design a framework that is able to exploit longer-term historical information from the object tracking algorithm to assist the object detection.


Scenario 1 - LiDAR point cloud sparsity issue

(a) Previous frame (b) Current Frame

Scenario 2 - Occlusion issue

(a) Previous frame (b) Current Frame

Fig. 1. Common problems of existing object detection frameworks. The green and red bounding boxes are the ground truth and inference results [36]. The first row shows object detection issues caused by point cloud sparsity and out-of-range. Even though vehicle No. 114 is detected and tracked in the previous frame, its bounding box varies significantly. Similarly, the vehicle No. 265 suffers from occlusion issues as can be seen it is not detected in the current frame.

To this end, we propose the HYbrid object Detection and tRacking for cOoperative perception framework named HYDRO-3D in this article to incorporate long-term historical information when performing object detection to further improve its resilience against the occlusion, sparsity of point cloud, and out-of-range issues in complex scenarios such as congested and out-of-range scenarios. What's more, how the historical information from the object tracking module benefits object detection has not been explored in both the object tracking by detection and joint object detection and tracking frameworks. Although the existing joint tracking and detection framework seeks to achieve both object detection and tracking functionality in only one network, it is difficult to investigate the implicit relationship between object detection and tracking. To answer this question, the HYDRO-3D is proposed to leverage the tracked cues of the objects to enhance object detection. The model explicitly forms a close loop between object detection and tracking, making them assist each other. The proposed model first extracts features from past tracked objects through a spatial-temporal pyramidal 3D network to assist object detection. The extracted abstract tracking features and detection backbone features will be later fused to infer objects in the LiDAR point cloud. After obtaining a set of detected objects, the model-based object tracking algorithm associates the detected objects, which are subsequently used to infer the objects in the next frame. During our experiments, not only can the bounding boxes of objects be predicted more

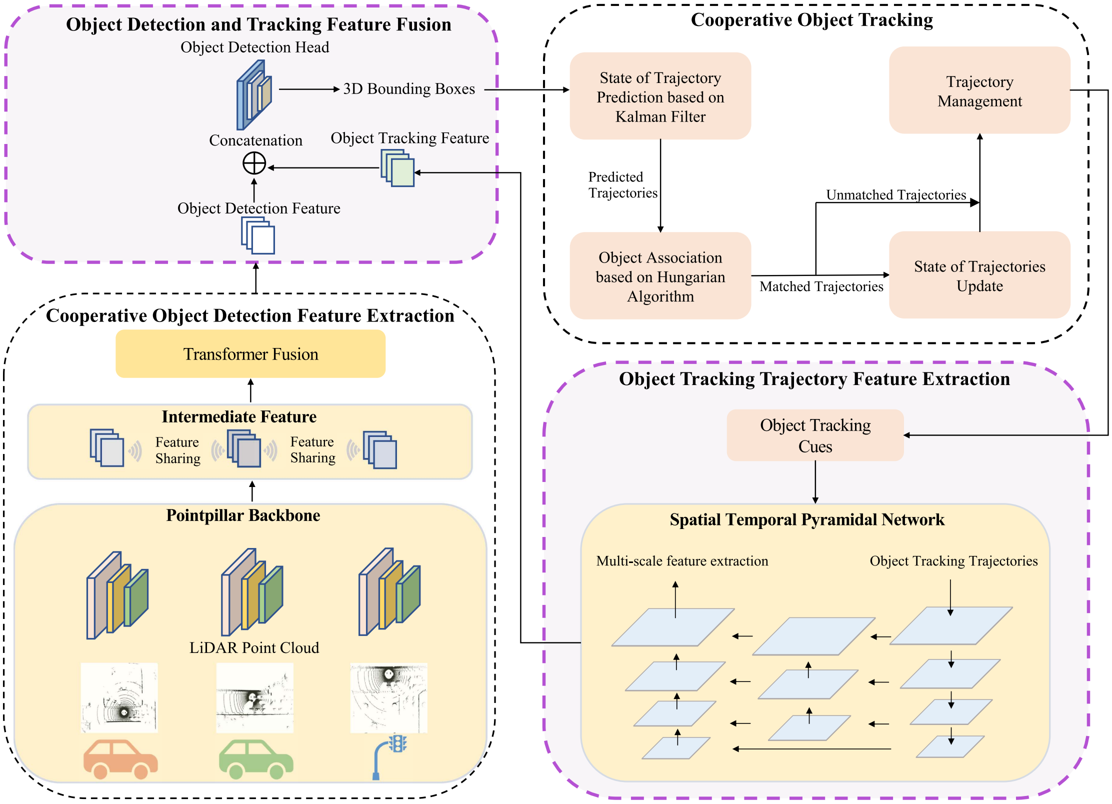

Object Detection and Tracking Feature Fusion
Object Detection Head
3D Bounding Boxes
Concatenation
Object Tracking Feature
Object Detection Feature
Cooperative Object Tracking
State of Trajectory Prediction based on Kalman Filter
Trajectory Management
Predicted Trajectories
Unmatched Trajectories
Object Association based on Hungarian Algorithm
Matched Trajectories
State of Trajectories Update
Cooperative Object Detection Feature Extraction
Transformer Fusion
Intermediate Feature
Feature Sharing
Feature Sharing
Pointpillar Backbone
LiDAR Point Cloud
Object Tracking Trajectory Feature Extraction
Object Tracking Cues
Spatial Temporal Pyramidal Network
Multi-scale feature extraction
Object Tracking Trajectories

Fig. 2. Hybrid object detection and tracking for cooperative perception (HYDRO-3D) framework. The HYDRO-3D consists of object detection feature extraction, object detection and tracking feature fusion, object tracking, and object tracking trajectory network (TTNET) modules. The PointPillar backbone [59] and intermediate feature fusion network (transformer) are used to extract the object detection features [36]. The object detection and tracking feature fusion module fuses these features and the object tracking features to predict objects as 3D bounding boxes. The object tracking algorithm will take the 3D bounding boxes and use a Kalman filter and a Hungarian algorithm to update and associate objects and their trajectories. The object-tracking trajectory network extracts multi-scale object-tracking features to enhance object detection.

accurately, but the number of erroneously detected objects is dramatically reduced. To summarize, the contributions of this work can be listed as follows:

- We propose HYDRO-3D, a novel cooperative object detection and tracking framework. In this framework, the historical object tracking information can be leveraged to assist the inference for object detection as shown in Fig. 2. Taking advantage of the objects' historical information, this framework will significantly improve object detection performance with short-term occlusion and out-of-range issues. In addition, the HYDRO-3D is a general object detection and tracking framework to be applied to both the cooperative perception and individual vehicle perceptions.
- In order to integrate global manipulations and low-level local operations from object tracking into object detection, we propose a spatial-temporal deep neural network shown by the object tracking trajectory network module in Fig. 2 to process the historical object tracking information. Then, we design a novel detection head to predict bounding boxes by fusing the heterogeneous object detection and tracking features.
- Based on the HYDRO-3D, we design a dedicated training strategy to train the networks in the framework easily. We further explicitly emphasize the benefits of incorporating object tracking into object detection, which is verified via comprehensive experiments and results.

The remainder of this article is organized as follows: Section II introduces our HYDRO-3D design including the overall architecture, cooperative object detection feature extraction, object detection and tracking feature fusion, the cooperative object tracking algorithm and then, the novel spatial-temporal network. Section III briefly describes our experimental setup and the quantitative and qualitative performance of the proposed HYDRO-3D for both single-vehicle and CAV scenarios. Section IV concludes this article.

II. Methodology

In this section, the architecture of HYDRO-3D in Section II-A is designed. Then, the object detection feature extraction is presented in Section II-B. The object detection and tracking feature fusion module is described in Section II-C. Next, the object tracking algorithm is detailed in Section II-D. The spatial-temporal neural network is detailed in Section II-E. The training strategy for the HYDRO-3D is illustrated in Section II-F.

A. Overall Architecture

Fig. 2 overviews the framework of the HYDRO-3D, which is comprised of object detection feature extraction, object detection and tracking feature fusion, object tracking, and object tracking trajectory network (TTNET) modules. The object detection feature extraction consists of a point cloud object detection (PointPillar [59]) backbone and a transformer intermediate feature fusion network to extract and fuse the object detection features from the adjacent CAVs [36]. In the object detection and tracking feature fusion module, the features from the object detection backbone and the object tracking features (historical information of the objects) are concatenated for the object detection head. With the concatenated features, the object detection head is able to generate 3D bounding boxes.

Upon obtaining the 3D bounding boxes for detected objects, the object tracking algorithm is to associate detected objects in the current frame with trajectories in the previous frames and provides their identification (ID) based on the Kalman filter algorithm [60] and the Hungarian method [61]. The main goal of Kalman filter is to predict the motion of detected objects based on their past tracklet and update the states of trajectories. Then the predicted trajectories and detected boxes are associated by the Hungarian method, which assigns the identity to each object. In addition, to improve the object detection performance with short-term occlusion and out-of-range issues, the historical trajectories of the objects from the object tracking module will be passed to the object tracking trajectory network (the spatial-temporal pyramidal network) and the tracking features will be obtained for object detection and tracking feature fusion.

Remark 1: As shown in Fig. 2, the object detection and tracking feature fusion and object tracking trajectory network highlighted in purple blocks, which are the two main contributions of this work, make this architecture HYDRO-3D a close loop from object detection and tracking feature fusion to object tracking. The TTNET serves as the bridge which allows us to leverage the historical objects' information to improve the object detection performance when fusing the object detection features and object tracking features from the temporal manner. In other words, through this close-loop framework, it is feasible to utilize the historical information to further assist object detection to address the occlusion, sparsity, and out-of-range issues for LiDAR-based object detection.

B. Cooperative Object Detection Feature Extraction

The HYDRO-3D builds on our previous work V2X-ViT [36], a state-of-the-art (SOTA) cooperative object detector. It identifies objects by using the shared feature-level information between smart infrastructure and CAVs. V2X-ViT first uses the PointPillar [59] backbone to extract features from the LiDAR point cloud in each agent and then utilizes the transformer to fuse the features from multiple agents. Specifically, the PointPillar backbone first calculates each pillar's attribute and then encodes pillars into pseudo images. Then the 2D convolution architecture could extract the point cloud features, which serve as the intermediate features for different agents. In order to fuse the extracted intermediate features, V2X-ViT computes two correlation scores, ATT and MSG, between these agents. Given different agents i and j, the ATT first computes the correlation scores between their intermediate features H_i and H_j. $||$ represents the concatenation operation. Similar to traditional attention, m represents the number of heads, $W^m_{key,j}$ and $W^m_{query,j}$ represent the Key and Query linear projector matrix [62].

$$K^m(j) = W^m_{k,c_j} \cdot H_j \tag{1}$$

$$Q^m(i) = W^m_{q,c_i} \cdot H_i \tag{2}$$

After computing the Key $K^m(j)$ and Query $Q^m(i)$, we calculate the attention score for each attention head $head^m_{ATT}(i,j)$. $W_{\theta(i,j)}$ is a linear projector matrix.

$$head^m_{ATT}(i,j) = \left(K^m(j) \cdot W^{m,ATT}_{\theta(i,j)}, Q^m(i)^T\right)/\sqrt{C} \tag{3}$$

$$ATT(i,j) = softmax\left(\underset{m\in[1,h]}{||}(head^m_{ATT})(i,j)\right) \tag{4}$$

Then, MSG is computed:

$$head^m_{MSG}(i,j) = K^m(j) \cdot H_j \cdot W^{m,MSG}_{i,j} \tag{5}$$

$$MSG(i,j) = \underset{m\in[1,h]}{||}(head^m_{MSG})(i,j) \tag{6}$$

After computing the ATT and the MSG scores, the intermediate feature H_i can be reconstructed as:

$$H_i = W_{c_i} \cdot ATT(i,j) \cdot MSG(i,j) \tag{7}$$

With H_i, V2X-ViT uses multi-scale window attention to improve the long-range spatial interaction.

Although V2X-ViT has already tried to address the occlusion, sparsity, and out-of-range issues of LiDAR point clouds by extending the perception range using shared LiDAR information from other CAVs, the performance of the object detection may still be compromised and the issue persists when the objects exist at the boundary of the shared LiDAR point clouds. To complement the performance of our research in V2X-ViT, we explore leveraging the object tracking historical trajectories information to enhance the object detection performance instead of merely relying on the current frame of LiDAR point clouds from different CAVs. As shown in Fig. 2, the object detection and tracking feature fusion module takes the separately extracted object detection and tracking features from the object detection feature extraction module and TTNET, respectively, and then fuses these separate features to infer the 3D bounding boxes of objects. More specifically, the object detection and tracking feature fusion module in HYDRO-3D further utilizes the intermediate features from the transformer encoder and the object tracking features from the TTNET to enhance object detection. This object detection and tracking feature fusion will be described in the subsequent Section II-C.

C. Object Detection and Tracking Feature Fusion (ODaTFF)

As mentioned in Section II-A, to tackle the occlusion, sparsity, and out-of-range issues of LiDAR-based object detection, we propose to leverage the object tracking features in the historical trajectories from the object tracking module to enhance the object detection performance. To this end, the object detection and tracking feature fusion module in HYDRO-3D will fuse both the object detection and tracking features when performing inferences for predicting 3D bounding boxes of objects.

Specifically, LiDAR point clouds and historical information can be exploited by combining the internal representation of object detection and tracking. As shown in Fig. 2, the object detection and tracking feature fusion module performs the intermediate fusion between the object detection and tracking feature maps. The concatenation operation, which combines the two feature tensors along their channel dimension, is used to fuse the two feature maps from the object detection and tracking and maintain their original features by (8). As shown in (9), the detection head is a 2D fully-connected layer followed by an activation function used to model the cross-module relations. The object tracking features serve as supplementary information to the detection head, which helps to detect distant and occluded objects with sparse or few points. While the detection head read the object detection features, the concatenation operation allows it to check the object tracking features to ensure the existence of objects from the historical trajectories and search for the geometric feature of objects, such as the shape and orientation.

$$f_{concat} = ||(f_{detection}, f_{tracking}) \tag{8}$$

$$f = \phi(W \cdot f_{concat} + b) \tag{9}$$

where the $||$ is the concatenation operation. The object detection feature $f_{detection}$ denotes the features from the cooperative object detection feature extraction module and $f_{tracking}$ represents the object tracking feature extracted from the cooperative object tracking module. W is the fully-connected layer. ϕ is a non-linear activation function.

Remark 2: It should be noted that the concatenation method used in this work is not unique and can be replaced by other dedicated fusion methods such as self-attention [63] and multi-model bilinear pooling [64], as long as they are capable to fuse the different or heterogeneous features for object detection. In other words, the focus of this work is to address the fusion problem between object detection and object tracking features with an appropriate algorithm, and thus, there may exist other better fusion solutions other than the concatenation approach used in this work and interested readers may try other alternatives. More importantly and generally, although the concatenation method itself is not a novel approach, the idea of fusing the information in the current LiDAR frame with the historical information from object tracking provides a new direction to both the traditional object tracking by detection and joint object detection and tracking frameworks.

D. Cooperative Object Tracking

After having the 3D bounding boxes of the objects from the object detection and tracking feature fusion module, a classical object tracking algorithm [45] in this subsection shown by the top right block in Fig. 2 is applied to associate the objects across different LiDAR frames to estimate the pose/trajectory of each object and provide the identity information.

1) Trajectory Prediction and Association: With the detected bounding boxes from the object detection and tracking feature fusion module in Fig. 2, all valid matches between detected bounding boxes $D(t)$ and trajectory $T(t-1)$ will need to be found, where t denotes the corresponding timestamp. To achieve this, a Kalman filter [65], [66] is applied to estimate the trajectories $T(t-1)$ of objects based on a constant velocity kinematic vehicle model. This estimated spatial information of trajectories combined with the information of detected objects, i.e., 3D bounding boxes, will be used to calculate the affinity matrix in the Hungarian algorithm to determine whether currently detected objects in $D(t)$ can be matched to trajectories in $T(t-1)$. Specifically, given a sequence of trajectories:

$$T(t-1) = \{T^j_{t-1}\}_{j=1}^{M_{t-1}} \tag{10}$$

at frame $t-1.j$ is the index of the corresponding bounding boxes. M_{t-1} is the previous associated trajectories. A constant velocity kinematic vehicle model in the Kalman filter is used to predict the position of the object in each trajectory in $T(t-1)$ as follows:

$$x^j_{t,pred} = x^j_{t-1} + v^j_{x_{t-1}} \tag{11}$$

$$y^j_{t,pred} = y^j_{t-1} + v^j_{y_{t-1}} \tag{12}$$

$$z^j_{t,pred} = z^j_{t-1} + v^j_{z_{t-1}} \tag{13}$$

where (x, y, z) correspond to the center of the bounding box. The final predicted trajectory is

$$T^j_{t,pred} = \Big(x^j_{t,pred}, y^j_{t,pred}, z^j_{t,pred}, \theta^j_{t-1}, w^j_{t-1}, \tag{14}$$

$$h^j_{t-1}, l^j_{t-1}, s^j_{t-1}, v^j_{x_{t-1}}, v^j_{y_{t-1}}, v^j_{z_{t-1}}\Big) \tag{15}$$

where subscript $pred$ means the variable is predicted by the Kalman filter, (w, h, l) represents the width, height, and length of the object, θ is the heading angle of the object, and s is the detection confidence score, which depends on the object detection and tracking feature fusion module. The additional variables (v_x, v_y, v_z) in trajectories represent the object velocity in x, y, and z directions.

After predicting the set of trajectories $T(t)_{pred}$, the 3D Intersection of Union (IoU) is used to compute the data affinity matrix $A \in \mathbf{R}^{M_{t-1} \times N_t}$ to determine the similarity between predicted trajectories and detected bounding boxes $D(t)$, where each element $A_{i,j}$ is the 3D IoU for the predicted trajectory i and the 3D bounding box j at frame t. The affinity matrix will be solved by the Hungarian algorithm, which considers the association as a bipartite matching problem to associate the corresponding objects.

2) State Update and Trajectory Management: After having the predicted trajectories and association results from the Hungarian algorithm, the Kalman Filter [67] is used to update the state of the predicted trajectory by considering the current detection objects and accounting for uncertainties from the detection errors. Accordingly, we have:

$$T^m_t = KF(T^m_t, D^k_t) \tag{16}$$

where $D^k_t \in D(t)$ and $T^m_t \in T(t)$ are the associated pair obtained from Hungarian algorithm, $k \in \{1, 2, \ldots, N_t\}$, $m \in \{1, 2, \ldots, M_t\}$. The updated state of corresponding predicted

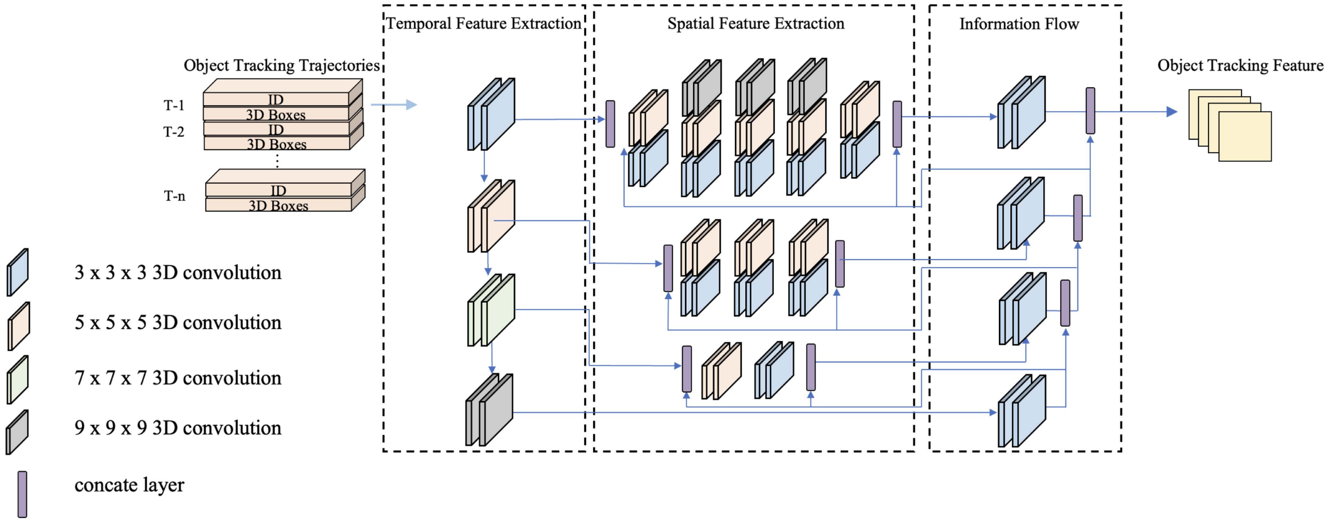


Fig. 3. The proposed spatial-temporal 3D network. The proposed TTNET is used to extract multi-scale object tracking features. Temporal feature extraction utilizes different convolution blocks to temporally extract multi-scale object tracking trajectories features. Then spatial feature extraction generates multiple level features. The information flow aggregates features from different levels to enhance both global and local feature extraction. Then the extracted object tracking features will be concatenated with the object detection features to predict the 3D bounding boxes.

trajectory $T_t^m \in T(t)$ is a weighted average between the related $D_t^k \in D(t)$ and $T_t^m \in T(t)$ [60].

Then, trajectory management will organize the new and old trajectories. When an object starts to appear at frame t, it could either be a false positive due to the detection error or it naturally enters the field of view. Similarly, when an object starts to disappear at frame t, it could either be a miss detection or it naturally leaves the LiDAR range. Both scenarios are handled by tracking objects in additional frames. Specifically, when D_t^l is an unmatched object entering the field of view, we will treat it as a new trajectory if D_t^l can be matched in the next few frames to prevent adding false positive detection as a new trajectory. When T_t^p is an unmatched trajectory leaving the field of view, we will treat it as a dead trajectory if T_t^p cannot be matched with any detected bounding boxes in the next couple of frames to prevent removing the true positive trajectory.

Remark 3: It is worth mentioning that although the trajectory management module in the object tracking algorithm will handle the birth and death of the corresponding object, it still highly relies on the performance of the object detection in particular in the area where the occlusion, sparsity, and out-of-range issues of LiDAR are severe. Better object detection around the area mentioned above will make better object tracking results. This motivates us to leverage the well-tracked objects' historical information to improve the performance of object detection as the actual objects won't appear and disappear immediately but the falsely detected objects may make this appearance and disappearance. If we can make full use the historical information as a memory, the sudden appearance, and disappearance of object detection can be avoided and therefore, the performance of object detection will be improved. To this end, we propose the object tracking trajectory network (TTNET) in the next subsection.

E. Object Tracking Trajectory Network (TTNET)

In order to provide the historical tracking objects' information for the ODaTFF module in Fig. 2, we propose a TTNET in this subsection. With this intention to provide linkage between the past object tracking cues and current object detection simultaneously, we propose a 3D-spatial-temporal pyramidal network, which propagates the global temporal information and local spatial information in the object tracking to the object detection. The temporal information records the historical movement paths of objects, the varying distance to the ego vehicle, and the number of occurrences of the same object, which is critical to determine the existence of objects in the current frame. The spatial information includes the 3D geometric information and locations of objects, which is crucial to keep the invariant size of the same object. Thus, the TTNET should perform both the global spatial operations, which judge the existence of objects, and the local spatial operations to transmit the geometric and location information of the same objects.

As shown in Fig. 3, the TTNET could process the object tracking cues at multi-scales through convolutional networks of different sizes. The base of the TTNET performs global operations through the large convolution blocks, which contain high-level features, such as the existence of objects. Similarly, the top of the TTNET generates low-level features through several small convolution blocks including the shape and position information of objects. All the global and local features will be further concatenated with the features obtained from the cooperative object detection feature extraction module in Fig. 2. Specifically, given the historical object tracking information, we first convert the 3D bounding boxes and ID to the anchor format. Then, we use a 3D convolutional neural network (CNN) [68] to perform the 3D convolutions shown by Fig. 3

over the spatial-temporal tracking anchor. As mentioned, the problem of mapping historical object tracking cues to the current detection involves both global and local manipulations. The global operations are used to find the high-level properties, such as the existence of objects, errors, or misses in the object tracking information, while the low-level processing is needed for the prediction of the specific position and orientation of objects. More importantly, there should be an interaction between the global and local modifications, as, for example, the existence of objects is critical for the coordinate predictions of objects. To address this issue, a novel 3D pyramidal CNN architecture, which processes the anchor at different scales and combines the learned global and local features together, allows the network to capture good features of bounding box information at smaller scales and performing mostly global anchor manipulations that are working with noisy object tracking information.

Fig. 3 illustrates the detailed schematic representation of the proposed deep learning architecture. The model has an inverted pyramidal shape and processes the bounding boxes from the object tracking algorithm at four different scales. The proposed architecture has a number of blocks that processes feature maps in parallel with 3D convolutions of different size, and the outputs of the corresponding convolutions layers are then concatenated, which allows the network to learn a more diverse set of features at each level. Maxpooling and ReLU activation function is applied after each convolution operation.

After discussing the detailed architecture of TTNET, we explain how TTNET leverages the tracking information to improve object detection performance. The TTNET model acts as a bridge between object detection and object tracking. The generated object-tracking features work as supplementary information to the object detection head in ODaTFF. With these features from TTNET, the detection head is able to judge the existence of objects based on both the object detection feature and the high-level object tracking feature, and at the same time uses the low-level features to correct the size of objects. From the perspective of feature map design, the feature maps generated by TTNET are in the same format as the detection features, which allows the object detection and tracking features of the same objects can correspond to the nearby regions in the feature maps. While performing the convolution operations, the detection head is able to fetch the same object features from the object detection and tracking feature maps. Thanks to prior knowledge provided by object tracking, the detection head can better predict whether the object exists, whether it will travel out of the perception range, and the geometric information about objects.

Remark 4: Compared to traditional object tracking by detection or joint-object-detection-tracking frameworks, it can be seen that the proposed TTNET allows us to take the object tracking information and generate the appropriate features as the additional prior knowledge about the objects to further assist the object detection. It differs from the object tracking-by-detection framework by incorporating the historical objects' information into the object detection and also distinguishes itself from the joint-object-detection-tracking framework in terms of two perspectives: 1) its capability to consider longer valuable historical objects' information when performing object detection; 2) the structure of the framework is straight forward and explainable from the functionality perspective compared with the object joint-detection-tracking framework. Benefiting from this clear structure, the improvement of object detection caused by historical object tracking information can be explicitly identified.

F. HYDRO-3D Training Strategy

As shown in Fig. 2, in our HYDRO-3D, there are three networks including object detection and tracking feature fusion, object tracking trajectory network, and V2X-ViT. The complexity of these three networks in our HYDRO-3D makes it difficult to train the HYDRO-3D. To properly train the HYDRO-3D, we adopted a dedicated training strategy instead of jointly training all the CNN modules in the HYDRO-3D.

Since the main objective of cooperative object detection and feature extraction is to extract corresponding features of objects and this module is relatively independent, our overall training logic is to separate the training of the cooperative object detection feature extraction module and the TTNET and let the detection head in ODaTFF make determinations based on the extracted object detection and tracking features. We adopted this strategy because first training overall HYDRO-3D will consume a lot of GPU memory as the point cloud features from CAVs occupy a lot of memories, putting the object tracking historical information together would exceed the GPU memories with desired batch size. Secondly, jointly training the cooperative object detection feature extraction module and TTNET modules would cause bias in object tracking features. During the training, when the historical tracking data are accurate, simply jointly training the HYDRO-3D from the scratch would cause the network to believe the object tracking features without obtaining satisfactory object detection features from the point clouds. This is because when the TTNET fetches accurate past tracking information, the detection head tends to directly utilize the object tracking features instead of learning from both object tracking and detection features. Therefore, the cooperative object detection feature extraction and TTNET are trained separately to ensure the corresponding features are well extracted. Then all the layers in the cooperative object detection feature extraction module are frozen and do not participate in the backpropagation. In the meantime, the TTNET module and ODaTFF module start to be trained. Finally, after training of cooperative object detection feature extraction and TTNET, the overall HYDRO-3D is jointly trained for a few more epochs. Such a training strategy not only lets both the object detection and tracking features be well extracted and not interfere with each other but also improves the training efficiency.

Remark 5: It should be noticed that the main goal of our training strategy is to extract well-trained features from each distinctive input without causing much bias to one of the features. Bear this in mind, such a training strategy can be generalized to other large complex multi-modality fusion tasks to extract the modality's features without causing feature bias problems when the network can fastly learn from one of the features.

III. Experiments

In this section, we first describe our experimental setup including datasets and implementation details. Then, we evaluate the quantitative and qualitative performance of the proposed HYDRO-3D on the LiDAR-based object detection tasks of CAV applications.

A. The Dataset

The experiments are built on our previous work, the large-scale cooperative perception dataset V2XSet [36], which is collected by using CARLA [69] and OpenCDA [14]. The number of intelligent agents that can communicate with each other ranges from two to seven across all sequences and could also vary over time within one sequence. These scenes often contain dense traffic, objects that are far away from the ego vehicle, and other diverse traffic scenarios where occlusion, sparsity, and out-of-range issues happen frequently and challenge object detection. The majority of our data comes from eight default towns provided by CARLA and we gather 55 representative scenes covering 5 different roadway types including the straight segment, curvy segment, midblock, entrance ramp, and intersection. The period of each scene is limited to 25 seconds. Each time step of the scene contains a single LiDAR point cloud frame and four RGB images. The training and validation split contains 6694 and 1920 LiDAR point cloud frames.

The LiDAR, global positioning system (GPS), and inertial measurement unit (IMU) data from the dataset are used for our experiments. The sensors are mounted on top of each CAV and infrastructure. The equipped LiDAR on each CAV has 32 channels of lasers and the range of the lasers is 120 meters. At the intersection, mid-block, and entrance, the infrastructure sensors are installed on the light poles at a height of 14 feet. The point clouds were recorded at 10 Hz and the corresponding GPS/IMU data and timestamp were saved. The ground-truth data contains the surrounding vehicle's 3D bounding boxes and their unique ID. More dataset details can be found in [36].

B. Implementation Details

To train the HYDRO-3D, information within 10 past object tracking frames along with the current LiDAR frame was used. The object tracking information contains both accurate and inaccurate information. The accurate information, which consists of the accurate bounding boxes and ID information, was directly obtained from the ground truth of V2XSet. The inaccurate bounding box came from the V2X-ViT output and inaccurate IDs were generated by the object tracking algorithm in Section II-D. The model was implemented in PyTorch and was trained on Nvidia V5000. Stochastic gradient descent with momentum [70] is used to train the model with a batch size of 12, a dropout of 0.5, a momentum of 0.9, and a learning rate of 0.01. Networks are trained for 80 epochs, and the learning rate is decayed by a factor of 10 at epoch 60. We initialize the spatial-temporal pyramid network with Kaiming initialization [71]. While for the Flow stream, we use stacks of 10 interleaved horizontal and vertical optical flow frames and use the kinetic model, provided by the authors of [41]. Both the V2X-ViT and our experiments are done under the perfect setting.

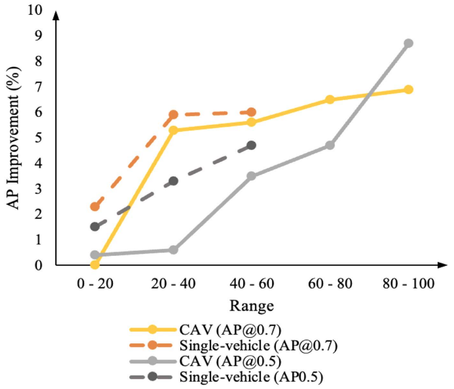


Fig. 4. The correlation between the object detection performance improvement and detection range. As the range increase, the object detection performance increases for both connected automated vehicles (CAVs) and single-vehicle.

C. Comprehensive Experimental Results

The HYDRO-3D is tested in the diverse scenes of the V2XSet dataset and the quantitative performance is discussed in this subsection. Table I shows the object detection comparisons between HYDRO-3D and V2X-ViT [36] on the V2XSet validation set. We report the results in the Average Precision [36] under the different thresholds. As the Table I shows, with the shared information from different CAVs, the object detection performance exceeds the V2X-ViT by 3.7% in AP@0.7. The performance improvement is positively correlated with the distance as shown in Fig. 4. When detecting objects at 40 - 60 meters, the detection performance has been increased by 5.6% in AP@0.7. The performance at 20 - 40 meters increased by 5.3%, which is slightly lower than the long-range improvement (40 - 60 meters). This is because when the vehicles are far away from the ego vehicle, the performance of detection will be impacted by the sparsity of the LiDAR point cloud. Compared to the V2X-ViT, our method can better detect these vehicles by leveraging the historical information of the tracked vehicles through the TTNET. However, for objects within 0 - 20 meters, there is no improvement. This is the advantage of using the shared information to achieve cooperative object detection in the original V2X-ViT because in this area, the point cloud is dense and the occlusion issue can easily be resolved by using the shared information to complement the point cloud of the ego vehicle. This can also be seen from Fig. 4 that, the improvement of object detection for single vehicle is higher than that of CAVs since the shared information can be used to resolve the challenges aforementioned to some extent.

As mentioned, because the shared information from other CAVs can naturally resolve the occlusion, sparsity, and out-of-range issues to some extent but the LiDAR-based single-vehicle object detection is prone to these issues, to further investigate the possible reasons for the variable improvement under different perception ranges, we also tested the model on the single vehicle

TABLE I
THE COMPARATIVE RESULTS OF OUR HYBRID OBJECT DETECTION AND TRACKING FOR COOPERATIVE PERCEPTION(HYDRO-3D) AND OTHER DETECTION MODELS ON THE V2X VALIDATION SCENARIOS. AS THE COOPERATIVE OBJECT DETECTION ON CAVS USES SHARED INFORMATION TO DETECT THE OBJECTS, THE SENSING RANGE CAN BE UP TO 100 M

CAV	AP@0.7 (Average)	AP@0.7 (0 - 20 m)	AP@0.7 (20 - 40 m)	AP@0.7 (40 - 60 m)	AP@0.7 (60 - 80 m)	AP@0.7 (80 - 100 m)
HYDRO-3D	**75.0**	**97.3**	**84.7**	**64.8**	**47.9**	**26.3**
V2X-ViT	71.3	**97.3**	79.4	59.2	41.4	19.4
	AP@0.5 (Average)	AP@0.5 (0 - 20 m)	AP@0.5 (20 - 40 m)	AP@0.5 (40 - 60 m)	AP@0.5 (60 - 80 m)	AP@0.5 (80 - 100 m)
HYDRO-3D	**89.1**	**98.9**	**95.0**	**84.7**	**73.3**	**52.3**
V2X-ViT	87.1	98.7	94.4	81.2	68.6	43.6

TABLE II
THE COMPARATIVE RESULTS OF OUR HYBRID OBJECT DETECTION AND TRACKING FOR COOPERATIVE PERCEPTION (HYDRO-3D) AND OTHER DETECTION MODELS ON THE SINGLE-VEHICLE VALIDATION SCENARIOS. DUE TO THE SPARSITY ISSUE OF POINT CLOUD, THE RANGE OF THE SINGLE VEHICLE LIDAR-BASED OBJECT DETECTION IS LIMITED TO ONLY 60 M

Single vehicle	AP@0.7 (Average)	AP@0.7 (0 - 20 m)	AP@0.7 (20 - 40 m)	AP@0.7 (40 - 60 m)
HYDRO-3D	**58.1**	**77.2**	**56.7**	**49.6**
V2X-ViT	53.6	74.9	50.8	43.4
	AP@0.5 (Average)	AP@0.5 (0 - 20 m)	AP@0.5 (20 - 40 m)	AP@0.5 (40 - 60 m)
HYDRO-3D	**68.7**	**81.9**	**68.2**	**66.4**
V2X-ViT	66.3	80.4	64.9	61.7

object detection and the results are listed in Table II. As demonstrated in Table II, our HYDRO-3D outperforms the original V2X-Vit by 5.9% in AP@0.7 for the single-vehicle detection at 20 - 40 meters. Nevertheless, when the objects are close to the ego vehicles at 0 - 20 meters, the performance of object detection improves 2.3% which is different from that of the CAV application. This is because although these detected objects are close to the ego vehicle, they are likely to be occluded such that V2X-Vit can not make proper inferences and the tracking cues, which are generated by the TTNET, are incorporated in the object detection and tracking feature fusion in HYDRO-3D and still help to detect these objects. Therefore, our HYDRO-3D framework is able to further enhance both cooperative object detection and single-vehicle object detection in terms of tackling occlusion, sparsity, and out-of-range issues.

Except for the quantitative performance of the HYDRO-3D on the validation set, some visualization regarding the typical scenarios is also provided to demonstrate how useful the historical information of objects from the object tracking is to assist object detection. The main advantages of leveraging object tracking to enhance object detection are for long-distance object detection and occluded object detection. First, as the first column in Fig. 5 shows, the HYDRO-3D outperforms V2X-ViT in terms of long-distance object detection because there is miss detection in V2X-ViT but the HYDRO-3D can predict the bounding box of the object normally. It is mainly because the vehicles that are far away from the ego vehicle are preserved in object tracking information, specifically, in the features generated by TTNET. In addition, the object detection performance for occluded vehicles can also be boosted by our HYDRO-3D as shown by the 2nd-4th columns in Fig. 5. The point cloud representation of these occluded vehicles is generally not complete/sufficient and sometimes sparse and thus, the object detection of V2X-ViT may fail. In our HYDRO-3D, the TTNET and ODaTFF can utilize historical information to make inferences to predict these occluded vehicles at the current LiDAR frame. What's more, the TTNET and ODaTFF in the HYDRO-3D can also prevent false positive errors because, in reality, vehicles have momentum and cannot suddenly appear or disappear. Therefore, when the object detection algorithm detects objects that suddenly appear in the current frame and don't exist in the features from TTNET, the ODaTFF has a low probability of outputting such vehicles as detected objects. After fusing both the object detection features and object tracking features, it is likely for ODaTFF to remove these false positive detected vehicles.

Therefore, in short, these comparisons between our HYDRO-3D and V2X-ViT further confirm the TTNET and object detection and tracking feature fusion modules using historical object tracking information for object detection will enhance the performance of long-distance object detection in particular where the sparsity of point cloud dominates the performance of object detection.

D. Ablation Study

To further investigate the effectiveness of leveraging the object tracking algorithm in Section II-D to the detection performance, we also replace the object tracking algorithm in Fig. 2 with the ground truth object tracking results for TTNET. Other modules Fig. 2 are the same. Then, the object detection results with ground truth object tracking historical information

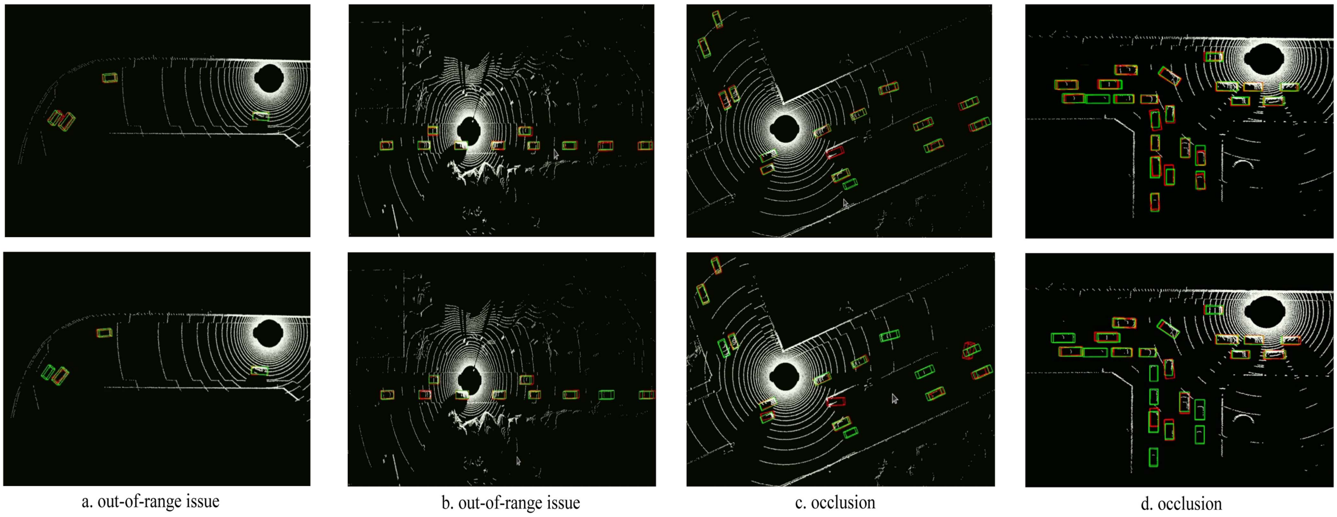


Fig. 5. The visualization of hybrid object detection and tracking for cooperative perception (HYDRO-3D). The first row gives the detection results of our HYDRO-3D. The second row is the results from V2X-ViT. The green bounding boxes represent the ground truth of the objects and the red ones are predicted by our HYDRO-3D and V2X-ViT. We compare the out-of-range and occlusion scenarios to illustrate our improvements.

TABLE III
THE PERFORMANCE OF THE HYBRID OBJECT DETECTION AND TRACKING FOR COOPERATIVE PERCEPTION ALGORITHM BASED ON THE ACCURATE OBJECT TRACKING INFORMATION. NOTE THAT OUR NOISY SETTING IS DIFFERENT FROM THE V2X-VIT [36] PAPER

CAV	AP@0.7 (Average)	AP@0.7 (0 - 20 m)	AP@0.7 (20 - 40 m)	AP@0.7 (40 - 60 m)	AP@0.7 (60 - 80 m)	AP@0.7 (80 - 100 m)
Noisy	75.0	97.3	85.7	66.8	47.9	26.3
Perfect	**75.7**	**97.3**	**85.9**	**67.6**	**49.0**	**27.7**
	AP@0.5 (Average)	AP@0.5 (0 - 20 m)	AP@0.5 (20 - 40 m)	AP@0.5 (40 - 60 m)	AP@0.5 (60 - 80 m)	AP@0.5 (80 - 100 m)
Noisy	89.1	98.9	95.0	84.7	73.3	52.3
Perfect	**89.9**	**98.9**	**95.3**	**85.3**	**74.3**	**52.5**

TABLE IV
THE COMPARATIVE RESULTS OF OUR HYBRID OBJECT DETECTION AND TRACKING FOR COOPERATIVE PERCEPTION (HYDRO-3D) AND OTHER DETECTION MODELS ON THE V2XSET DATASET. NOTE THAT OUR NOISY SETTING IS DIFFERENT FROM THE V2X-VIT [36] PAPER

Single vehicle	AP@0.7 (Average)	AP@0.7 (0 - 20 m)	AP@0.7 (20 - 40 m)	AP@0.7 (40 - 60 m)
Noisy	57.6	77.2	56.7	49.6
Perfect	**58.5**	**77.2**	**56.9**	**50.5**
	AP@0.5 (Average)	AP@0.5 (0 - 20 m)	AP@0.5 (20 - 40 m)	AP@0.5 (40 - 60 m)
Noisy	68.7	81.9	68.2	66.4
Perfect	**69.9**	**81.9**	**68.7**	**68.1**

in the loop can be gathered. The object detection results with accurate object tracking information and inaccurate real object tracking information are listed in Tables III and IV, respectively. The "perfect" in the first column means that the HYDRO-3D takes the ground truth of object tracking for the ODaTFF to make inferences. The "noisy" represents the results taking the output of the object tracking algorithm Fig. 2 as inaccurate object tracking information. From Tables IV and III, we can observe the influence of the performance of object tracking on object detection: The more accurate object tracking results are, the better object detection performance is for both cooperative object detection in Table IV and single vehicle object detection Table III. However, it is also worth mentioning that, although having more accurate object-tracking information does result in more accurate object detection in our HYDRO-3D, the improvement is not that substantial. On another aspect, based on Table IV Table III, it can be inferred that the object tracking algorithm Fig. 2 is valid for our holistic framework of HYDRO-3D and contribute to the performance improvement of object detection through the TTNET and ODaTFF.

IV. Conclusion

In this article, a novel hybrid object detection and tracking for cooperative perception framework is presented to leverage the object tracking historical information further enhance object detection in areas where occlusion, sparsity, and out-of-range issues affect object detection heavily. To achieve this, a spatial-temporal pyramidal 3D network in the TTNET is designed to generate the object tracking features. With these features, the object detection feature and tracking feature fusion module is put forward to fuse the current object detection features and the historical object tracking features to make robust inferences of objects. The following conclusions can be drawn based on the comprehensive experimental results: 1) our HYDRO-3D is capable of incorporating the historical object tracking information to assist object detection; 2) the object tracking information can benefit the object detection for both cooperative object detection for CAVs and object detection for the individual vehicle; 3) better object tracking information in our HYDRO-3D framework makes better object detection performance. In detail, our HYDRO-3D has outperformed the SOTA V2X-ViT by 3.7% and 4.5% in AP@0.7 of object detection for CAV and single-vehicle, respectively.

References

[1] S. Nallamothu et al., "Detailed concept of operations: Transportation systems management and operations/cooperative driving automation use cases and scenarios," Federal Highway Admin., Washington, D.C., USA, Tech. Rep. FHWA-HRT-20-064, 2020.

[2] X. Xia et al., "An automated driving systems data acquisition and analytics platform," *Transp. Res. Part C: Emerg. Technol.*, vol. 151, 2023, Art. no. 104120.

[3] R. Xu et al., "The OpenCDA open-source ecosystem for cooperative driving automation research," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 2698–2711, Apr. 2023.

[4] W. Liu, L. Xiong, X. Xia, Y. Lu, L. Gao, and S. Song, "Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model," *IET Intell. Transp. Syst.*, vol. 14, no. 10, pp. 1183–1189, 2020.

[5] Y. Dai, L. Zhang, K. Yan, Q. Chen, and Z. Zhou, "An integrated co-operative control strategy for EVs accessed community uninterruptible power system," *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2482–2493, Mar. 2023.

[6] L. Chen, Y. Zhang, B. Tian, Y. Ai, D. Cao, and F.-Y. Wang, "Parallel driving OS: A ubiquitous operating system for autonomous driving in CPSS," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 886–895, Dec. 2022.

[7] V. K. Yanumula, P. Typaldos, D. Troullinos, M. Malekzadeh, I. Papamichail, and M. Papageorgiou, "Optimal trajectory planning for connected and automated vehicles in lane-free traffic with vehicle nudging," *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2385–2399, Mar. 2023.

[8] A. Abdelhalim and M. Abbas, "A real-time safety-based optimal velocity model," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 165–175, 2022.

[9] L. Chen et al., "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1046–1056, Feb. 2023.

[10] Y. Zhang, W. Wu, and W. Zhang, "Noncooperative game-based cooperative maneuvering of intelligent surface vehicles via accelerated learning-based neural predictors," *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2212–2221, Mar. 2023.

[11] M. Hua et al., "Energy management of multi-mode plug-in hybrid electric vehicle using multi-agent deep reinforcement learning," 2023, *arXiv:2303.09658*.

[12] W. Liu, X. Xia, L. Xiong, Y. Lu, L. Gao, and Z. Yu, "Automated vehicle sideslip angle estimation considering signal measurement characteristic," *IEEE Sens. J.*, vol. 21, no. 19, pp. 21675–21687, Oct. 2021.

[13] T. Wang et al., "UMC: A unified bandwidth-efficient and multi-resolution based collaborative perception framework," 2023, *arXiv:2303.12400*.

[14] R. Xu, H. Xiang, X. Xia, X. Han, J. Liu, and J. Ma, "OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proc. Int. Conf. Robot. Automat.*, 2021, pp. 2583–2589.

[15] E. Thonhofer et al., "Infrastructure-based digital twins for cooperative, connected, automated driving and smart road services," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 311–324, 2023.

[16] S. E. Shladover, "Opportunities and challenges in cooperative road vehicle automation," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 216–224, 2021.

[17] A. Coppola, L. Di Costanzo, L. Pariota, and G. N. Bifulco, "Fuzzy-based variable speed limits system under connected vehicle environment: A simulation-based case study in the city of Naples," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 267–278, 2023.

[18] M. Hua, G. Chen, B. Zhang, and Y. Huang, "A hierarchical energy efficiency optimization control strategy for distributed drive electric vehicles," *Proc. Inst. Mech. Engineers, Part D: J. Automobile Eng.*, vol. 233, no. 3, pp. 605–621, 2019.

[19] H. Xie, Y. Wang, X. Su, S. Wang, and L. Wang, "Safe driving model based on V2V vehicle communication," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 449–457, 2022.

[20] M. Cebecauer, W. Burghout, E. Jenelius, T. Babicheva, and D. Leffler, "Integrating demand responsive services into public transport disruption management," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 24–36, 2021.

[21] L. Chen, Q. Ding, Q. Zou, Z. Chen, and L. Li, "DenseLightNet: A light-weight vehicle detection network for autonomous driving," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10600–10609, Dec. 2020.

[22] M. Schutera, M. Hussein, J. Abhau, R. Mikut, and M. Reischl, "Night-to-day: Online image-to-image translation for object detection within autonomous driving by night," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 480–489, Sep. 2021.

[23] M. Hu, X. Wang, Y. Bian, D. Cao, and H. Wang, "Disturbance observer-based cooperative control of vehicle platoons subject to mismatched disturbance," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 2748–2758, Apr. 2023.

[24] S. Teng et al., "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Trans. Intell. Veh.*, vol. 8, no. 6, pp. 3692–3711, Jun. 2023.

[25] C. Vitale, P. Kolios, and G. Ellinas, "Optimizing vehicle re-ordering events in coordinated autonomous intersection crossings under CAVs' location uncertainty," *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3473–3488, May 2023.

[26] Y. Guo and J. Ma, "Leveraging existing high-occupancy vehicle lanes for mixed-autonomy traffic management with emerging connected automated vehicle applications," *Transportmetrica A: Transport Sci.*, vol. 16, no. 3, pp. 1375–1399, 2020.

[27] K. Raboy, J. Ma, E. Leslie, and F. Zhou, "A proof-of-concept field experiment on cooperative lane change maneuvers using a prototype connected automated vehicle testing platform," *J. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 77–92, 2021.

[28] J. Ma, E. Leslie, A. Ghiasi, Z. Huang, and Y. Guo, "Empirical analysis of a freeway bundled connected-and-automated vehicle application using experimental data," *J. Transp. Eng., Part A: Syst.*, vol. 146, no. 6, 2020, Art. no. 04020034.

[29] M. Razzaghpour, R. Valiente, M. Zaman, and Y. P. Fallah, "Predictive model-based and control-aware communication strategies for cooperative adaptive cruise control," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 232–243, 2023.

[30] M. Kloock, P. Scheffe, O. Gress, and B. Alrifaee, "An architecture for experiments in connected and automated vehicles," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 175–186, 2023.

[31] J. Li et al., "Learning for vehicle-to-vehicle cooperative perception under lossy communication," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 2650–2660, Apr. 2023.

[32] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, "Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 397–410, 2022.

[33] E. Andreotti, Selpi, and M. Aramrattana, "Cooperative merging strategy between connected autonomous vehicles in mixed traffic," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 825–837, 2022.

[34] K. Yang, C. A. Haddad, G. Yannis, and C. Antoniou, "Classification and evaluation of driving behavior safety levels: A driving simulation study," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 111–125, 2022.

[35] Y. Guo, J. Ma, E. Leslie, and Z. Huang, "Evaluating the effectiveness of integrated connected automated vehicle applications applied to freeway managed lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 522–536, Jan. 2022.

[36] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, “V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer,” in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 107–124.

[37] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3464–3468.

[38] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3D traffic scene understanding from movable platforms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1012–1025, May 2014.

[39] G. Li, Z. Ji, X. Qu, R. Zhou, and D. Cao, “Cross-domain object detection for autonomous driving: A stepwise domain adaptative YOLO approach,” *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 603–615, Sep. 2022.

[40] L. Chen et al., “Surrounding vehicle detection using an FPGA panoramic camera and deep CNNs,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 5110–5122, Dec. 2020.

[41] T. Zhou, J. Chen, Y. Shi, K. Jiang, M. Yang, and D. Yang, “Bridging the view disparity between radar and camera features for multi-modal fusion 3D object detection,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1523–1535, Feb. 2023.

[42] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 770–779.

[43] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.

[44] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2018, pp. 4490–4499.

[45] X. Weng, J. Wang, D. Held, and K. Kitani, “3d multi-object tracking: A baseline and new evaluation metrics,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10359–10366.

[46] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, “Probabilistic 3D multi-object tracking for autonomous driving,” 2020, *arXiv:2001.05673*.

[47] M.-Q. Dao and V. Frémont, “A two-stage data association approach for 3D multi-object tracking,” *Sensors*, vol. 21, no. 9, 2021, Art. no. 2894.

[48] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, “3D multi-object tracking in point clouds based on prediction confidence-guided data association,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5668–5677, Jun. 2022.

[49] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3D object detection and tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2021, pp. 1190–1199.

[50] A. Kim, A. Ošep, and L. Leal-Taixé, “EagerMOT: 3D multi-object tracking via sensor fusion,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11315–11321.

[51] 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*, 2012, pp. 3354–3361.

[52] H. Caesar et al., “nuScenes: A multimodal dataset for autonomous driving,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2020, pp. 11618–11628.

[53] J. Kini, A. Mian, and M. Shah, “3DMODT: Attention-guided affinities for joint detection & tracking in 3D point clouds,” 2022, *arXiv:2211.00746*.

[54] J. Willes, C. Reading, and S. L. Waslander, “InterTrack: Interaction transformer for 3D multi-object tracking,” 2022, *arXiv:2208.08041*.

[55] H. Liu, Y. Ma, Q. Hu, and Y. Guo, “CenterTube: Tracking multiple 3D objects with 4D tubelets in dynamic point clouds,” *IEEE Trans. Multimedia*, early access, Feb. 1, 2023, doi: 10.1109/TMM.2023.3241548.

[56] S. Wang, Y. Sun, C. Liu, and M. Liu, “PointTrackNet: An end-to-end network for 3-D object detection and tracking from point clouds,” *IEEE Robot. Autom.*, vol. 5, no. 2, pp. 3206–3212, Apr. 2020.

[57] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, “Infrastructure-based object detection and tracking for cooperative driving automation: A survey,” in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 1366–1373.

[58] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, and K. Oguchi, “PillarGrid: Deep learning-based cooperative perception for 3D object detection from onboard-roadside LiDAR,” in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 1743–1749.

[59] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 12697–12705.

[60] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Fluids Eng.*, vol. 82, pp. 35–45, 1960.

[61] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Nav. Res. Logistics Quart.*, vol. 2, no. 1–2, pp. 83–97, 1955.

[62] D. Zhang, F. Zhou, Y. Jiang, and Z. Fu, “MM-BSN: Self-supervised image denoising for real-world with multi-mask based on blind-spot network,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2023, pp. 4188–4197.

[63] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 464–468.

[64] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visual grounding,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 457–468.

[65] X. Xia, E. Hashemi, L. Xiong, and A. Khajepour, “Autonomous vehicle kinematics and dynamics synthesis for sideslip angle estimation based on consensus kalman filter,” *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 1, pp. 179–192, Jan. 2023.

[66] X. Xia et al., “Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors,” *Mech. Syst. Signal Process.*, vol. 162, 2022, Art. no. 107993.

[67] L. Gao, L. Xiong, X. Xia, Y. Lu, Z. Yu, and A. Khajepour, “Improved vehicle localization using on-board sensors and vehicle lateral velocity,” *IEEE Sens. J.*, vol. 22, no. 7, pp. 6818–6831, Apr. 2022.

[68] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2018, pp. 6450–6459.

[69] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[70] W. Liu, K. Quijano, and M. M. Crawford, “YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8085–8094, 2022.

[71] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.