A Cooperative Vehicle Localization and Trajectory Prediction Framework Based on Belief Propagation and Transformer Model

Feiyu Jin, Kai Liu, *Senior Member, IEEE*, Chunhui Liu, Tongtong Cheng, Hao Zhang, *Member, IEEE*, and Victor C. S. Lee, *Member, IEEE*

***Abstract*—The advancement of sensing, transmission, and computation technologies has transformed modern vehicles into ubiquitous consumer electronics. This paper presents a cooperative vehicle localization and trajectory prediction framework for enabling Intelligent Transportation Systems (ITS) applications. Specifically, the proposed framework consists of a Belief Propagation based Location Approximation (BPLA) algorithm for cooperative vehicle localization and a Transformer-based Vehicle trajectory prediction model called VFormer. The BPLA algorithm first establishes a factor graph based on the sensor measurements transmitted by vehicles, and then adopts a modified belief propagation procedure to approximate the posterior distribution of vehicles. On this basis, VFormer extracts the hidden features from historical positions estimated by BPLA and vehicle motion data to model long-term and short-term motion patterns of vehicles. Moreover, the multi-head attention layer in the VFormer is utilized to learn the spatial-temporal dependencies between the target vehicle and its surrounding vehicles at different timestamps to improve prediction accuracy. A comprehensive performance evaluation has been conducted based on the public vehicle trajectory dataset and the real-world system prototype. Experiment results demonstrate the superiority of the proposed framework on improving vehicle localization and trajectory prediction performance.**

***Index Terms*—Vehicle localization, trajectory prediction, belief propagation, transformer model, Internet of Vehicles.**

I. Introduction

RECENT advances in wireless communication, multi-modal sensing and big data processing technologies drive the development of intelligent vehicles, which have emerged as modern consumer electronics [1]. A variety of consumer-level ITS applications, such as collision avoidance, intersection navigation and autonomous driving are developing rapidly, which requires the reliable estimation of vehicles' locations and the accurate prediction of their trajectories [2]. Global Navigation Satellite System (GNSS) is widely adopted for vehicle localization, however, the localization accuracy cannot be satisfied for most safety-critical applications solely based on GNSS in complex traffic environments. Great efforts have been devoted to improving the localization accuracy for GNSS [3], including Real Time Kinematic (RTK) correction [4], multi-sensor fusion [5] and trajectory map matching [6]. Nonetheless, none of them have considered the cooperation of vehicles on localization.

Several studies have focused on cooperative vehicle localization based on sharing information through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) links in vehicular communications [7]. Initially, radio-based ranging techniques were utilized to measure inter-vehicle distances such as Time Of Arrival (TOA) [8], Received Signal Strength (RSS) [9] and Doppler shift [10] to facilitate the cooperative localization, but these techniques faced limitations in terms of ranging accuracy. Nowadays, vehicles are equipped with advanced sensors such as Light Detection and Ranging (LiDAR), which can achieve ranging accuracy within a few centimeters under favorable conditions. Consequently, Various data fusion methods [11], [12], [13], [14] have been investigated to enhance the precision of cooperative localization through the utilization of these advanced sensors.

Vehicle trajectory prediction has also received considerable attention in the past decade. Early efforts include physical-based and maneuver-based methods, which relies on certain motion models or identified vehicles' maneuvers for prediction. Recently, with the remarkable success of Deep Learning (DL) and the efficient deployment of DL models on resource-constrained mobile devices [15], researchers have increasingly turned to design deep sequential models, such as Recurrent Neural Networks (RNN) [16] and Long Short-Term Memory networks (LSTM) [17] for trajectory prediction. In order to further model interactions among vehicles, various techniques including graph neural network [18], social pooling [19] and attention mechanisms [20], [21] are integrated into the deep sequential models to enhance the prediction performance.

One significant challenge for practical vehicle trajectory prediction in real-world scenarios is that current prediction

Manuscript received 28 September 2023; revised 15 November 2023 and 30 December 2023; accepted 5 February 2024. Date of publication 8 February 2024; date of current version 26 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62172064, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN202100637. *(Corresponding author: Kai Liu.)*

Feiyu Jin, Kai Liu, Chunhui Liu, and Tongtong Cheng are with the College of Computer Science, Chongqing University, Chongqing 400040, China (e-mail: fyjin@cqu.edu.cn; liukai0807@cqu.edu.cn; chhliu0302@cqu.edu.cn; totcheng0105@stu.cqu.edu.cn).

Hao Zhang is with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: zhanghao@cqupt.edu.cn).

Victor C. S. Lee is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: csvlee@eee.hku.hk).

Digital Object Identifier 10.1109/TCE.2024.3364052

models tend to perform well only on well-established datasets, where vehicle trajectories are captured with a high degree of precision. However, the ground truth of a vehicle's position is often unavailable in real-world settings, and the location provided by GNSS is prone to errors. This leads to a notable discrepancy between practical implementation and theoretical research. Furthermore, prediction models tend to be more effective in forecasting the relative displacements rather than absolute locations, which need to be closely integrated with the positioning system to yield meaningful prediction results. On the other hand, current vehicle localization methods should also be improved to satisfy the requirement of trajectory prediction models. Existing methods primarily concentrate on enhancing the accuracy of the current locations, without taking into account the optimization of previously recorded locations to form a more precise trajectory.

With above motivations, this work is dedicated to designing and implementing a tightly coupled cooperative localization and trajectory prediction framework based Internet of Vehicles (IoV) to fulfill the practical requirements of various ITS applications. Specifically, the proposed framework consists of two methods to realize the cooperative vehicle localization and trajectory prediction, namely, Belief Propagation based Location Approximation (BPLA) algorithm for cooperative localization and a Transformer [22] based model named VFormer for predicting future positions of vehicles. Within this framework, all the measurements collected by vehicles are aggregated at the Roadside Unit (RSU). Then, the BPLA algorithm is executed based on collected data to determine the locations of different vehicles simultaneously and further refine the historical localization results for trajectory prediction. Utilizing the enhanced trajectories provided by BPLA, coupled with motion data from inertial sensors, the VFormer model is capable of predicting the future positions of each vehicle, effectively accounting for the inherent noise in practical measurements. The main contributions of this work are listed as follows:

- We propose a BPLA algorithm to effectively infer the posterior distribution of all anticipated vehicles. First, a factor graph is firstly established to model the relationship between fused sensor measurements and vehicle location distribution, where location distribution of vehicles are considered as the variables and sensor measurements including GNSS results, inter-vehicle distance are modeled as factors connected to variables. Next, a modified belief propagation procedure is designed to estimate the posterior distribution of each variable through a dedicated two-phase belief propagation. In the forward phase, the vehicle's location can be quickly determined through local belief propagation. In the backward phase, the historical locations can be further refined based on belief back propagation through time.
- We propose a VFormer model to achieve robust trajectory prediction in the presence of noisy location measurements. VFormer model takes historical positions and vehicle motion as input features, which are initially embedded into higher-dimensional features to facilitate the learning of long-term and short-term vehicle dynamics for prediction, compensating the impact of localization errors in the historical trajectory. Moreover, a spatial-temporal learning mechanism is designed, which flatten the input features from surrounding vehicles into a long sequence, enabling the multi-head attention layer to capture dependencies across the target vehicle and its surrounding vehicles at different timestamps. Finally, a generative style decoder is implemented to predict the future trajectories of vehicles at one forward process, thereby improving the inference efficiency of the prediction model.
- We conduct a comprehensive evaluation of the proposed framework through both simulation studies and field testing. The simulation is conducted based on a public naturalistic vehicle trajectory dataset. Further, we implement the system prototype with four-wheel robots on the campus. The results demonstrate that BPLA can significantly enhance the localization accuracy compared to GNSS. Furthermore, VFormer outperforms other baseline models, even when considering varying levels of localization errors and prediction intervals.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the system architecture. Section IV and Section V propose the BPLA algorithm and VFormer model, respectively. Section VI gives a performance evaluation based on the public trajectory dataset and real-world system prototype. Finally, Section VII concludes this work.

II. Related Work

A. Vehicle Localization

A significant amount of research has been devoted to vehicular localization. Traditional approaches mainly focused on overcoming the limitations of GNSS with auxiliary information, such as onboard sensors, vehicle kinematics, and digital road maps, etc. With the development of vehicular networks, cooperative localization among vehicles, which benefits from vehicle-to-vehicle interactions, has been increasingly investigated. Liu et al. [23] proposed to improve localization accuracy by using GNSS pseudo-range measurements to detect inter-vehicle distances and then proposed a Distributed Location Estimate Algorithm (DLEA) to realize cooperative localization based on non-linear optimization. Cruz et al. [9] leveraged the RSS in V2V communication to measure the distances between vehicles based on some fitted Path loss model. On this basis, a particle filter was proposed to determine the vehicle's position based on RSS measurements, inertial sensor data and GNSS results of different vehicles. Liu et al. [10] utilized the range-rating capability of Dedicated Short-Range Communication (DSRC) for intervehicle distance measurement and proposed GNSS/DSRC integration method for cooperative localization, in which a Huber M-estimation technique is introduced for cubature Kalman filter to improve the robustness of the data fusion.

Recently, more powerful sensors such as LiDAR and high-resolution cameras have been deployed on the roadside and vehicles. Many studies were conducted to fully utilize these advanced sensors to improve the performance of cooperative localization. Fang et al. [12] considered correlation and outliers of various data sources and proposed an iterated split covariance intersection filter based cooperative localization method in a decentralized framework to address these two types of error data sources simultaneously. Yang et al. [13] proposed a multi-sensor multi-vehicle localization framework, which consists of a local filter (e.g., Kalman filter, particle filter) to estimate the vehicle's self-state and the states of its neighbor based on sensor data and a global filter to solve the data fusion as an optimization problem, aiming to find a optimal weight for linear combination of estimation from neighbors. Soatti et al. [11] proposed an ICP framework, in which vehicles can collectively determine the positions of surrounding static environment features (traffic lights, pedestrians, etc.) and use them as reference points to further refine their own position. In order to solve the Data Association in ICP (i.e., associate the observed passive features into corresponding sensed objects), Brambilla et al. [24] further proposed a method named Implicit Cooperative Positioning with Data Association (ICP-DA), which jointly solves the DA issue and cooperative localization based on distributed belief propagation. To address computational complexity of cooperative localization in decentralized inference frameworks, Kim et al. [14] established a Constraint Satisfaction Problem (CSP) based optimization formulation for cooperative localization and developed a three-step algorithmic framework to organize the message flow and scheduling for an efficient distributed solution.

B. Vehicle Trajectory Prediction

Great efforts have been made for vehicle trajectory prediction. Traditional approaches include physic-based and maneuver-based methods, where the former assumes vehicles' movement follows a certain motion model and the latter identifies vehicles' maneuvers for prediction. On this basis, some works combine physic-based and maneuver-based methods to improve performance based on some weighting function [25] or Interactive Multiple Models (IMM) [26].

Nowadays, Neural Networks such as RNN and LSTM have been developed for trajectory prediction and achieve satisfactory performance. Zyner et al. [16] proposed a multi-modal probabilistic solution for driver intention and path prediction, in which RNN combined with a mixture density output layer is firstly adopted and the output of the network is then passed through a clustering algorithm to produce a ranked set of possible trajectories. Xing et al. [27] proposed a personalized trajectory prediction for the leading vehicle, in which different driving styles are first recognized based on the Gaussian Mixture Model (GMM). Then, a joint time series model based on LSTM and RNN is proposed to predict future trajectories by incorporating different driving styles.

Spatial-temporal relationship among vehicles on the road has been exploited for vehicle trajectory prediction. Deo and Trivedi [19] proposed an LSTM encoder-decoder model with convolutional social pooling (CS-LSTM) for robustly learning interdependencies in vehicle motion. In addition, a maneuver-based LSTM decoder is designed in the proposed model, which can output a multi-modal predictive distribution over future trajectories based on maneuver classes. Wang et al. [28] proposed an LSTM encoder-decoder framework integrated with Generative Adversarial Networks (GANs) for trajectory prediction, where an auto-encoder convolution mechanism and a recurrent social mechanism are utilized to model spatial and temporal relations among multiple vehicles, respectively. While, Sheng et al. [18] tackle spatial interactions among vehicles using a Graph Convolutional Network (GCN) and utilize a convolution neural network to capture temporal features. Then, the spatial-temporal features are encoded and decoded by a Gated Recurrent Unit (GRU) network to generate future trajectory distributions.

Several works have adopted attention mechanisms for trajectory prediction. Lin et al. [21] proposed a spatio-temporal attention LSTM model (STA-LSTM), in which the temporal-level attention identifies important historical trajectories and the spatial-level attention can rank neighboring vehicles in terms of their influences on the target vehicle. Similarly, Messaoud et al. [20] adopted the multi-head attention mechanism to derive the relative influence of surrounding vehicles with respect to their future motion.

Distinguish from previous works, in the proposed framework, BPLA can simultaneously optimize vehicle localization by two-phase belief propagation process within a factor graph, enhancing both current and historical localization accuracy. VFormer further extends beyond existing models by considering both current and past states of neighbor vehicles when modelling the interactions among vehicles.

III. System Architecture

In this section, we present the system architecture of IoV-based cooperative vehicle localization and trajectory prediction, as illustrated in Fig. 1. In the considered scenario, each vehicle is equipped with an Onboard Unit (OBU) for vehicular communication and three different types of sensors including a GNSS receiver, motion-related sensors and ranging sensors (e.g., RADAR, LiDAR, camera-based detector). Vehicles within the communication radius of RSU transmit all onboard sensor measurement data to the RSU. Given the relatively small data size of these measurements, the transmission delay is ignored in the current framework. However, when transmitting large sensor data such as raw point clouds and high-resolution pictures, the network reliability [29] and the transmission delay should be considered to guarantee an efficient data distribution [30].

The GNSS localization results of each vehicle are considered as coarser measurements of the vehicle's current location. The motion data from the inertial sensors (e.g., accelerometer, speedometer) is recorded during the vehicle's driving, enabling the computation of the vehicle's movement through kinematic models. Furthermore, the inter-vehicle distance of nearby vehicles is measured by equipped ranging devices. Following the collection of inter-vehicle measurements by the RSU, the initial step of data fusion involves aligning the timestamps of

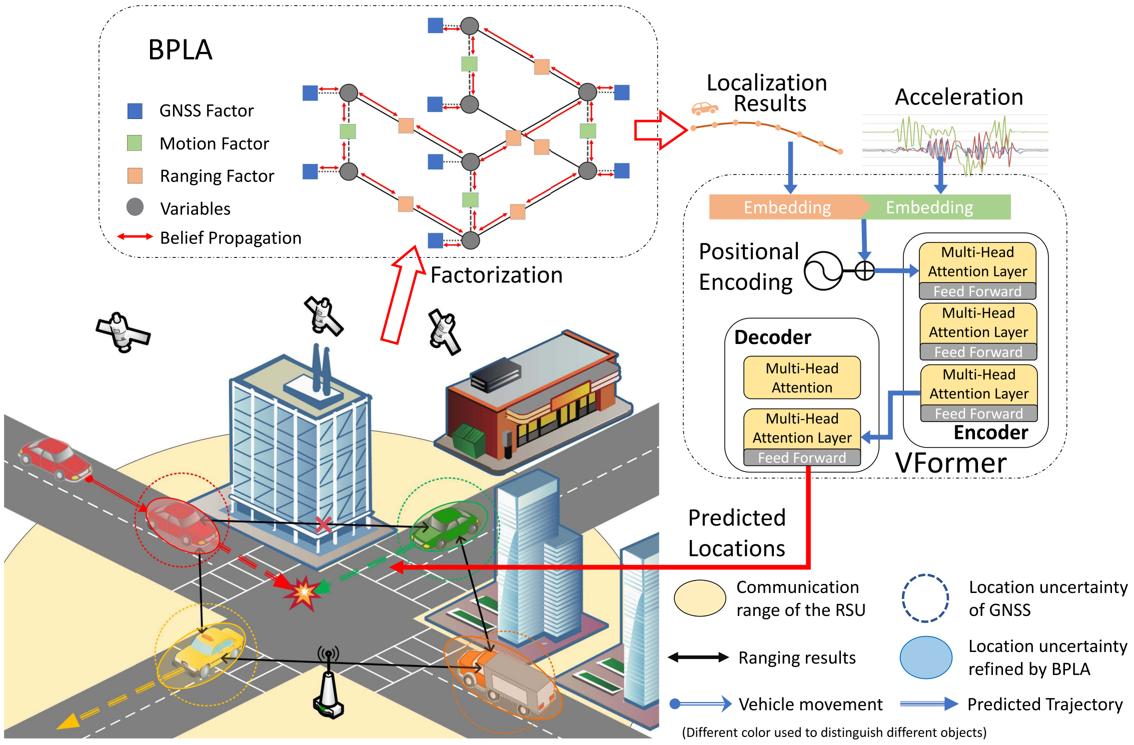


Fig. 1. The framework of cooperative vehicle localization and trajectory prediction.

measurements originating from various sensors and distinct vehicles. Subsequently, the RSU associates each measurement with its respective source and target vehicles and calculates a singular inter-vehicle distance between two vehicles capable of mutual detection. In the BPLA algorithm, the joint probability distribution of vehicles' locations, along with the three types of measurements, is factorized within the factor graph. Dedicated belief propagation for this factor graph reduces the localization uncertainty of GNSS results, allowing the RSU to achieve a more accurate understanding of vehicles' location on the road. Building on this, the refined historical trajectories from the past few seconds, combined with the inertial sensor readings, are fed into VFormer to predict the vehicle's future location over a short period.

With these information, RSU is capable of identifying potential collisions on the road even when two vehicles are under Non-Line-of-Sight (NLOS) conditions, as shown in Fig. 1. Except safety-critical applications, some other primary functions of ITS including traffic scheduling, IoV networking can be also benefited from the proposed frameworks. Therefore, we argue that the proposed framework can be easily extended to existing ITS infrastructure and provide critical information for various ITS applications to boost their performance.

IV. Belief Propagation Based Vehicle Localization

In this section, we propose the Belief Propagation based Location Approximation (BPLA) algorithm. In BPLA, we first construct a factor graph based on the fused sensor measurements. Next, we design a modified belief propagation procedure to effectively approximate the posterior distribution of each vehicle's location based on an iterative message passing on the constructed factor graph.

A. Factor Graph Construction

Consider a set of vehicles $V_t = \{v_{1,t}, v_{2,t}, \ldots, v_{|N_t|,t}\}$ is on the target area deployed at timestamp t, where $|N_t|$ is total number of vehicles at timestamp t. The positions of the i-th vehicle up to time t are denoted by $\boldsymbol{l}_{i,1:t}^{(\mathcal{V})} = \{\boldsymbol{l}_{i,1}^{(\mathcal{V})}, \boldsymbol{l}_{i,2}^{(\mathcal{V})}, \ldots, \boldsymbol{l}_{i,t}^{(\mathcal{V})}\}$, where $\boldsymbol{l}_{i,t}^{(\mathcal{V})} = [\, x_{i,t}^{(\mathcal{V})} \; y_{i,t}^{(\mathcal{V})} \,]^\top$ representing the coordinate of vehicle position.

Each vehicle has 3 types of sensors, namely, GNSS receiver, inertial sensor and ranging sensor. The GNSS measurement of the i-th vehicle at time t is denoted by $\boldsymbol{l}_{i,t}^{(G)} = [\, x_{i,t}^{(G)} \; y_{i,t}^{(G)} \,]^\top$. The inertial sensors including accelerometer, odometer and gyroscope monitor the vehicles motion state continuously, which can be represented by $\boldsymbol{m}_{i,t} = [\mathrm{v}_{i,t}^{(x)}, \mathrm{v}_{i,t}^{(y)}, a_{i,t}^{(x)}, a_{i,t}^{(y)}]$, where $\mathrm{v}_{i,t}^{(x)}, \mathrm{v}_{i,t}^{(y)}, a_{i,t}^{(x)}, a_{i,t}^{(y)}$ are velocity and acceleration along X axis and Y axis at time t, respectively. Moreover, the vehicle v_i can detect its distance to nearby vehicles within the detection range R_s and the set of vehicle pairs that can detect each other is denoted by $E_t = \{(v_i, v_j) \in V_t \times V_t \mid \|\boldsymbol{l}_{i,t}^{(\mathcal{V})} - \boldsymbol{l}_{j,t}^{(\mathcal{V})}\| < R_s\}$, and the corresponding ranging measurements set is defined as $\boldsymbol{D}_t = \{d_t(i,j) \;\mid (v_{i,t}, v_{j,t}) \in E_t\}$, where $d_t(i,j) = [\, \Delta x_{ij} \; \Delta y_{ij} \,]^\top$ denote the relative distance detected by the ranging sensor between v_i and v_j at X and Y axis. Based on above definitions, the set of all the sensor measurements at timestamp t is $Z_t = \{\boldsymbol{l}_{i,t}^{(G)}, \boldsymbol{m}_{i,t}, \boldsymbol{D}_t \mid v_i \in V_t\}$. The primary notations are summarized in Table I.

TABLE I
SUMMARY OF PRIMARY NOTATIONS

Notations	Description
$\boldsymbol{l}^{(\mathcal{V})}_{i,t}$	i-th vehicles' true location at timestamp t
$\boldsymbol{l}^{(\mathcal{G})}_{i,t}$	i-th vehicles' GNSS result at timestamp t
$\boldsymbol{m}_{i,t}$	i-th vehicles' inertial sensor measurements at timestamp t
$d_t(i,j)$	ranging result between i and j-th vehicle at timestamp t
$f^{(g)}_{i,t}$	GNSS measurement factor
$f^{(m)}_{i,k-1\Rightarrow k}$	vehicle state transition factor
$f^{(d)}_{i\leftrightarrow j,t}$	inter-vehicle distance measurement factor
$\boldsymbol{l}^{(\mathcal{O})}_{i,t}$	location measurement for i-th vehicle at timestamp t
$\alpha_{i,t}$	acceleration measurement for i-th vehicle at timestamp t
$\boldsymbol{l}^{(\mathcal{R})}_{i,t}$	relative displacement for i-th vehicle at timestamp t
$\Delta\boldsymbol{l}_{i,t+f}$	future displacement for i-th vehicle at timestamp $t+f$

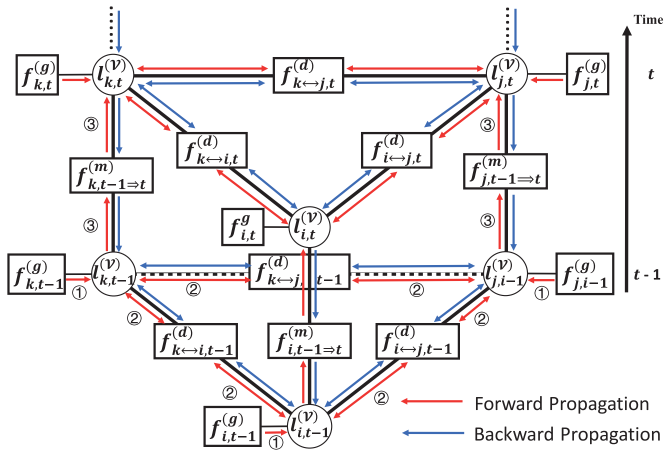


Fig. 2. Factor graph.

Consider each vehicle's position at time t as a two dimensional probability distribution $P(\boldsymbol{l}^{(\mathcal{V})}_{1,t})$. The probability distribution of all vehicles from the initial timestamp to current timestamp t conditioning on the sensor measurement set $Z_{1:t}$ is denoted by $P(\boldsymbol{l}^{(\mathcal{V})}_{1,1:t}, \boldsymbol{l}^{(\mathcal{V})}_{2,1:t}, \ldots\ldots, \boldsymbol{l}^{(\mathcal{V})}_{|N_t|,1:t} \mid Z_{1:t})$ where $Z_{1:t} = \{Z_1, Z_2, \ldots, Z_t\}$, and the goal of cooperative localization is therefore to determine the marginal distribution of each vehicle v_i at each timestamp $P(\boldsymbol{l}^{(\mathcal{V})}_{i,t})$.

To effectively calculate the marginal distribution of each vehicle up to timestamp $\hat{t}$, the joint probability distribution can be factorized based on conditional independence of the different measurements, which is calculated as follows:

$$P\Big(\boldsymbol{l}^{(\mathcal{V})}_{1,1:\hat{t}}, \boldsymbol{l}^{(\mathcal{V})}_{2,1:\hat{t}}, \ldots\ldots, \boldsymbol{l}^{(V)}_{|N_{\hat{t}}|1:\hat{t}}\Big) \propto \prod_{t=1}^{\hat{t}} \prod_{v_i\in V_t} f^{(g)}_{i,t} f^{(m)}_{i,t\Rightarrow t+1} \prod_{(v_i,v_j)\in E_t} f^{(d)}_{i\leftrightarrow j,t} \tag{1}$$

where $f^{(g)}_{i,t} = f(\boldsymbol{l}^{(\mathcal{V})}_{i,k})$ is the factor related to GNSS measurements, $f^{(m)}_{i,t\Rightarrow t+1} = f(\boldsymbol{l}^{(\mathcal{V})}_{i,t}, \boldsymbol{l}^{(\mathcal{V})}_{i,t+1})$ is the factor modelling the state transition from timestamp t to $t+1$ and $f^{(d)}_{i\leftrightarrow j,t} = f(\boldsymbol{l}^{(\mathcal{V})}_{i,k}, \boldsymbol{l}^{(\mathcal{V})}_{j,k})$ models the inter-vehicle distance measurement. Based on the factorization, we can establish a Factor Graph to represent the joint probability distribution. Factor graph [31] is a probabilistic graph model consisting of two types of nodes, namely, variable and factor. Variable nodes represent the random variables in the distribution and factor nodes define the relationships between variables in the graph. Fig. 2 presents a scenario where 3 vehicles can detect each other. As shown, each variable is linked to 3 different factors. $f^{(g)}_{i,t}$ connects the variable at timestamp t, representing the GNSS results for i-th vehicle at that time. $f^{(m)}_{i,t-1\Rightarrow t}$ connects the variable at timestamp $t-1$ to variable at timestamp t, modeling the state transition of i-th vehicle during this period. $f^{(d)}_{i\leftrightarrow j,t}$ connects two variables that can detect each other to reflect their distance relationship.

We model the three types of factors as Gaussian distribution and adopt the information form to describe the three different factors. The factor related to GNSS measurement $f^{(g)}_{i,t}$ is defined by:

$$f^{(g)}_{i,t} = \eta \exp\left\{-\frac{1}{2}\boldsymbol{l}^{(\mathcal{V})\top}_{i,t}\Omega_g \boldsymbol{l}^{(\mathcal{V})}_{i,t} + \xi_g^\top \boldsymbol{l}^{(\mathcal{V})}_{i,t}\right\} \tag{2}$$

where Ω_g is the precision matrix, which is the inverse of covariance matrix Σ_g of GNSS measurement, i.e., $\Omega_g = \Sigma_g^{-1}$. ξ_g is information vector, which is calculated by $\xi_g = \Omega_g \boldsymbol{l}^{(G)}_{i,t}$. The factor $f^{(d)}_{i\leftrightarrow j,t}$ related to the inter-vehicle distance measurement between i-th and j-th vehicle at timestamp t is defined by

$$f^{(d)}_{i\leftrightarrow j,t} = \eta \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{j,t}\end{bmatrix}^\top \Omega_d \begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{j,t}\end{bmatrix} + \xi_d^\top \begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{j,t}\end{bmatrix}\right\} \tag{3}$$

Given the covariance matrix $\Sigma_d = \begin{bmatrix}\sigma_d^2 & 0\\ 0 & \sigma_d^2\end{bmatrix}$ for distance measurement $d_t(i,j)$, the information vector ξ_d and precision matrix Ω_d can be computed by:

$$\begin{aligned}\Omega_d &= \mathrm{J}_s^\top \Sigma_d^{-1}\mathrm{J}_s\\ \xi_d &= \mathrm{J}_s^\top \Sigma_d^{-1} d_t(i,j)\end{aligned} \tag{4}$$

where $\mathrm{J}_s = \begin{bmatrix}-1 & 0 & 1 & 0\\ 0 & -1 & 0 & 1\end{bmatrix}$. Similarly, the factor $f^{(m)}_{i,t-1\Rightarrow t}$ related to state transition from time $t-1$ to t is defined by

$$f^{(m)}_{i,t-1\Rightarrow t} = \eta \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{i,t-1}\end{bmatrix}^\top \Omega_m \begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{i,t-1}\end{bmatrix} + \xi_m^\top \begin{bmatrix}\boldsymbol{l}^{(\mathcal{V})}_{i,t}\\ \boldsymbol{l}^{(\mathcal{V})}_{i-1,t}\end{bmatrix}\right\} \tag{5}$$

The corresponding covariance matrix for the displacement measurement is defined as $\Sigma_m = \begin{bmatrix}\sigma_m^2 & 0\\ 0 & \sigma_m^2\end{bmatrix}$. On this basis, the information vector ξ_m and precision matrix Ω_m is computed by

$$\Omega_m = \mathrm{J}_s^\top \Sigma_m^{-1}\mathrm{J}_s$$

$$\xi_m = \mathrm{J}_s^\top \Sigma_m^{-1} \begin{bmatrix} v_{i,t-1}^{(x)}\Delta t + \frac{1}{2}a_{i,t-1}^{(x)}\Delta t^2 \\ v_{i,t-1}^{(y)}\Delta t + \frac{1}{2}a_{i,t-1}^{(y)}\Delta t^2 \end{bmatrix} \tag{6}$$

where Δt is the time interval between timestamp t and timestamp $t-1$. The covariance matrix $\Sigma_g, \Sigma_d, \Sigma_m$ used in these factors represents the uncertainty of measurements, which are adjustable parameters in the proposed algorithm.

B. Modified Belief Propagation

Since all the factors and variables in the factor graph are modeled as Gaussian distribution, we resort to Gaussian Belief Propagation [32] (GBP) to realize effective inference on the established factor graph. Building upon the standard GBP algorithm, we have divided the belief propagation into two phases, i.e., forward propagation and backward propagation. The forward propagation can quickly estimate the vehicle's location based on solely newly collected sensor measurements. While, the backward propagation can further refine the vehicles' historical trajectories by propagating the belief from variables with newly updated beliefs to their corresponding connected variables at the previous timestamps.

Forward Propagation:

Step 1. Initialization of variables belief with GNSS measurement factor: Since variables in the factor graph are initialized as a zero-mean Gaussian distribution, we first send messages from its related GNSS measurement factor to update its belief, as shown in Fig. 2. GNSS measurement factors only connect to the corresponding variable, its message to the variable is the factor itself, which is denoted by

$$\mu_{\left[f_{i,t}^{(g)} \to \boldsymbol{l}_{i,t}^{(\mathcal{V})}\right]} = f_{i,t}^{(g)} \tag{7}$$

Step 2. Variables belief updating by exchanging messages through distance measurement factor: After collecting messages from GNSS measurement factor, each variable in the current timestamp sends its messages to neighbor factors, which is calculated as follows:

$$\mu_{\left[\boldsymbol{l}_{j,t}^{(\mathcal{V})} \to f_{i\leftrightarrow j,t}^{(d)}\right]} = \prod_{l \in ne\left(\boldsymbol{l}_{j,t}^{(\mathcal{V})}\right) - f_{i\leftrightarrow j,t}^{(d)}} \mu_{\left[f_l \to \boldsymbol{l}_{j,t}^{(\mathcal{V})}\right]} \tag{8}$$

where $ne(\boldsymbol{l}_{j,t}^{(\mathcal{V})})$ is neighbor factor set of variables $\boldsymbol{l}_{j,t}^{(\mathcal{V})}$. Then, distance measurement factor sends messages to neighbor variables based on the incoming messages, which is calculated as follow:

$$\mu_{\left[f_{i\leftrightarrow j,t}^{(d)} \to \boldsymbol{l}_{i,t}^{(\mathcal{V})}\right]} = \int f_{i\leftrightarrow j,t}^{(d)} \mu_{\left[\boldsymbol{l}_{j,t}^{(\mathcal{V})} \to f_{i\leftrightarrow j,t}^{(d)}\right]} d\boldsymbol{l}_{j,t}^{(\mathcal{V})} \tag{9}$$

After collecting messages sent from its all-neighbor factors, the belief of $\boldsymbol{l}_{i,t}^{(\mathcal{V})}$ can be updated by:

$$b\left(\boldsymbol{l}_{i,t}^{(\mathcal{V})}\right) = \prod_{l \in ne\left(\boldsymbol{l}_{i,t}^{(\mathcal{V})}\right)} \mu_{f_l \to \boldsymbol{l}_{j,t}^{(\mathcal{V})}} \tag{10}$$

As shown in Fig. 2, each variable sends and receives messages with neighbor factors in **step 2**. Through multiple rounds of propagation, inter-vehicle distance measurements can be fused with GNSS measurements to realize a more accurate estimation of vehicles' location.

Step 3. Forward belief propagation to variables at new timestamp: After vehicles have moved for a certain period, the new sensor measurements are collected for the new position of vehicles. As shown in Fig. 2, we can add a new set of variables for the factor graph and connect them to corresponding variables at the last timestamp with transition factors and then send messages through transition factors to provide extra location information for new variables. Similarly, the message sent from the last variable to the state transition factor is calculated as

$$\mu_{\left[\boldsymbol{l}_{j,t-1}^{(\mathcal{V})} \to f_{j,t-1\Rightarrow t}^{(m)}\right]} = \prod_{l \in ne\left(\boldsymbol{l}_{j,t-1}^{(\mathcal{V})}\right) - f_{j,t-1\Rightarrow t}^{(m)}} \mu_{\left[f_l \to \boldsymbol{l}_{j,t-1}^{(\mathcal{V})}\right]} \tag{11}$$

Then, the transition factor can send its message to new variables, which is computed by:

$$\mu_{\left[f_{j,t-1\Rightarrow t}^{(m)} \to \boldsymbol{l}_{j,t}^{(\mathcal{V})}\right]} = \int f_{j,t-1\Rightarrow t}^{(m)} \mu_{\left[\boldsymbol{l}_{j,t-1}^{(\mathcal{V})} \to f_{j,t-1\Rightarrow t}^{(m)}\right]} d\boldsymbol{l}_{j,t-1}^{(\mathcal{V})} \tag{12}$$

Finally, the **step 2** are repeatedly applied for variables at the new timestamp to calculate their belief.

***Backward Propagation*:** Note that the number of vehicles varies at different timestamps, the fusion results can be more accurate when the number of vehicles is larger. Therefore, we further propagate Gaussian message backward to enhance the estimation accuracy for historical vehicle locations so that the input of the prediction model can be more reliable. Messages are sent from variables at timestamp t to corresponding transition factor after their belief is updated, which is denoted by $\mu_{\boldsymbol{l}_{i,t}^{(\mathcal{V})} \to f_{i,t\Rightarrow t-1}^{(m)}}$ and then messages from transition factor are further sent to variables at timestamp $t-1$, which is denoted by $\mu_{f_{i,t\Rightarrow t-1}^{(m)} \to \boldsymbol{l}_{i,t-1}^{(\mathcal{V})}}$. Then, we update the variable's belief at the last timestamp $t-1$ based on **step 2** at forward propagation. Such procedures can be repeated for several previous timestamps to further improve the localization accuracy of these variables.

V. Transformer Based Vehicle Trajectory Prediction Model

In this section, we propose a Transformer based vehicle trajectory prediction model based on the observed historical locations and sensor measurements collected from vehicles. In the following, we first define the basic inputs and output of the proposed models. Then, we introduce the architecture of VFormer with some dedicated designs to improve trajectory prediction performance.

A. Model Inputs and Outputs

Due to the unavailability of ground truth positions of vehicles in real-world scenarios, a set of observed locations of i-th vehicle for past k seconds is defined as $\mathcal{T}_{i,t-k:t}^{(\mathcal{O})} = \{\boldsymbol{l}_{i,t}^{(\mathcal{O})}, \boldsymbol{l}_{i,t-2}^{(\mathcal{O})}, \ldots, \boldsymbol{l}_{i,t-k}^{(\mathcal{O})}\}$ and each position is two-tuple $\boldsymbol{l}_{i,t}^{(\mathcal{O})} = (x_{i,t}^{(\mathcal{O})}, y_{i,t}^{(\mathcal{O})})$ representing the coordinate at X and Y axis, respectively. Besides, the acceleration of the vehicle is also recorded during the same period, which is denoted by $\alpha_{i,t-k:t} = \{\boldsymbol{a}_{i,t}, \boldsymbol{a}_{i,t-1}, \ldots, \boldsymbol{a}_{i,t-k}\}$, where $\boldsymbol{a}_{i,t} = (a_{i,t}^{(x)}, a_{i,t}^{(y)})$ represents the acceleration along X and Y axis. In addition, the relative displacements between historical

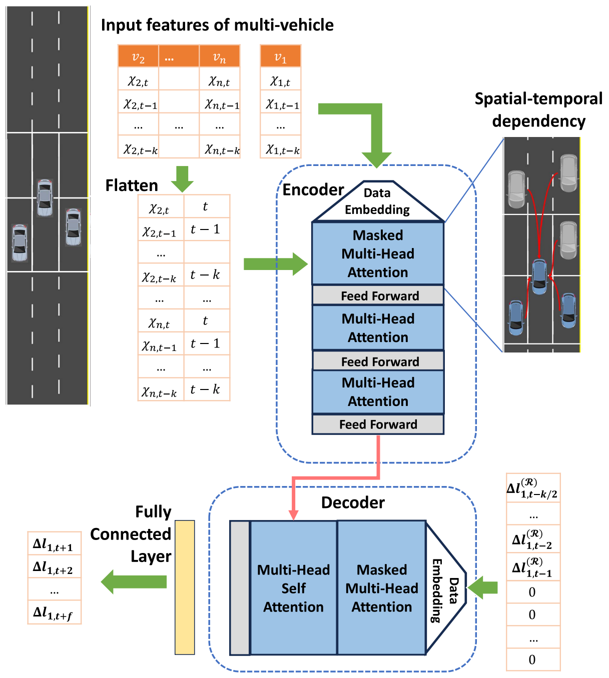


Fig. 3. Architecture of VFormer.

positions and current vehicle position are also explicitly calculated as an input feature, which is denoted by $\Delta l^{(\mathcal{R})}_{i,t-k:t} = \{\Delta \boldsymbol{l}^{(\mathcal{R})}_{i,t}, \Delta \boldsymbol{l}^{(\mathcal{R})}_{i,t-2}, \ldots, \Delta \boldsymbol{l}^{(\mathcal{R})}_{i,t-k}\}$, where $\Delta \boldsymbol{l}^{(\mathcal{R})}_{i,t-j} = (x^{(\mathcal{O})}_{i,t-j} - x^{(\mathcal{O})}_{i,t}, y^{(\mathcal{O})}_{i,t-j} - y^{(\mathcal{O})}_{i,t})$. These three types of data can represent the state of the vehicle at a given timestamp, which is denoted by $\chi_{i,t-k} = (\boldsymbol{a}_{i,t-k}, \boldsymbol{l}^{(\mathcal{O})}_{i,t-k}, \Delta \boldsymbol{l}^{(\mathcal{R})}_{i,t-k})$. Therefore, we adopt the historical states during a certain period as the model input for trajectory prediction.

When predicting the future trajectory of i-th vehicle, the historical states of its surrounding vehicles should also be considered, since other vehicles' driving routes could significantly affect the target vehicle's future trajectory. More specifically, as shown in Fig. 3, we assume target vehicle can observe vehicles within $\pm L$ meters longitudinally and two adjacent lanes laterally. These vehicles are considered as neighbors for the target vehicle. Suppose there are total N vehicles within the area, the input sequence for the target vehicle and its neighbor vehicle is denoted by $\boldsymbol{X} = \{\chi_{i,j} \mid i \in [1, N+1], j \in [t-k, t]\}$. With such input sequences, the goal of VFormer is to predict the corresponding displacement of the target vehicle $\boldsymbol{Y} = \Delta l_{i,t+1:t+f} = \{\Delta \boldsymbol{l}_{i,t+1}, \Delta \boldsymbol{l}_{i,t+2}, \ldots, \Delta \boldsymbol{l}_{i,t+f}\}$ for future f seconds. Suppose the ground truth position for i-th vehicle at current timestamp t is $\boldsymbol{l}_{i,t}$, where $\boldsymbol{l}_{i,t} = (x_{i,t}, y_{i,t})$. The displacement for i-th vehicle between the current position and the position at future f second is computed by $\Delta \boldsymbol{l}_{i,t+f} = (x_{i,t+f} - x_{i,t}, y_{i,t+f} - y_{i,t})$.

B. Network Architecture of VFormer

As depicted in Fig. 3, in contrast to conventional methods that solely rely on previous location measurements to model the long-term spatial context of the trajectory, VFormer leverages accelerations to infer the short-term changes in vehicle dynamics, which can benefit the prediction performance and compensate for the noisy location measurements. Therefore, a dedicated data embedding procedure is designed in VFormer to embed the input sequences of three types of data into higher dimension features, enabling the model to effectively learn the long-term and short-term movement patterns of the target vehicle.

Then, the embedded feature is processed through the encoder in the VFormer model. VFormer has 3 layers in the encoder, each encoder layer is identical to the layer used in the encoder of Transformer [22] model, consisting of a multi-head attention layer and a position-wise feed-forward layer. A residual connection is employed around each of the two sub-layers, followed by layer normalization. In the encoder, we design a spatial-temporal learning mechanism for modeling dependencies between the target vehicle and its surrounding vehicles at different timestamps based on a masked multi-head attention layer. Then, we stack two encoder layers to comprehensively learn the temporal dependencies of vehicles based on different attention heads. Finally, a generative style decoder proposed in [33] is deployed to generate the future positions of the target vehicle with a single forward process, avoiding time-consuming "dynamic decoding". In our setting, relative displacements in the last half of the input sequence are utilized as the start token in the decoder. We employ the same embedding method for the decoder inputs and then stack two multi-head attention layers to model dependencies, with the final one interacting with the output of the encoder to infer the final results. The dedicated design of the data embedding procedure and spatial-temporal learning in the VFormer model are introduced as follows:

Data Embedding: For three types of data in the input sequences, we employ independent linear layers with the activation function to embed each type of data into a $d/3$-dimensional vector and ultimately concatenate them into a d-dimensional vector to represent the input features, where d is feature dimension in VFormer. On this basis, we adopt the sinusoidal embedding method utilized in the original Transformer model to encode the positional information. A similar embedding procedure is employed for the input sequences of surrounding vehicles. According to the aforementioned definitions, there are total N surrounding vehicles and the length of the input sequence is k, which forms an input matrix of shape (N, k) as shown in Fig. 3. We flatten the input data of surrounding vehicles into an extended sequence $(N * k, 1)$ serving as both keys and values in multi-head attention. Consequently, owing to the flattening of the time sequences of multiple vehicles into a unified extended sequence, we need to replicate the positional embedding for each vehicle's input sequence.

Spatial-Temporal learning based on Multi-head Attention: For the first encoder layer, the embedded features of the target vehicle are considered as queries in the multi-head attention, while the embedded features of surrounding vehicles are considered as the keys and values. Therefore, each feature of the target vehicle will in turn query the states of all N neighbor vehicles for previous k timestamps in the flatten sequence. This gives the model a chance to learn not only the influence of surrounding vehicles on the target vehicle at the same timestamp t but also the influence of preceding states of surrounding vehicles from timestamp $t-k$ to t. Note that masks should be applied to attention scores to ensure that the embedded feature of the target vehicle at timestamp t does not extract information from embedded features of other vehicles with timestamps exceeding t.

More specifically, in the encoder layer, the linear transform is first adopted for the embedded feature $\mathbf{h}_i^{(\tau)}$ of i-th states of the target vehicle in input sequence to calculate the query $\mathbf{Q}_i$. Another two linear transforms are conducted for the embedded feature $\mathbf{h}_j^{(s)}$ in the flatten sequence of surrounding vehicle, where $j \in [1, N*k]$. Therefore, the key $\mathbf{K}_i$, query $\mathbf{Q}_j$ and value $\mathbf{V}_j$ are computed by:

$$\begin{aligned}\mathbf{Q}_i &= \mathbf{W}_{\boldsymbol{q}}\mathbf{h}_i^{(\tau)}\\ \mathbf{K}_j &= \mathbf{W}_{\boldsymbol{k}}\mathbf{h}_j^{(s)}\\ \mathbf{V}_j &= \mathbf{W}_{\boldsymbol{v}}\mathbf{h}_j^{(s)}\end{aligned} \tag{13}$$

where $\mathbf{W}_{\boldsymbol{q}}, \mathbf{W}_{\boldsymbol{k}}, \mathbf{W}_{\boldsymbol{v}}$ are learnable weights in linear transform for calculating the key, query and value component, respectively. Next, we need to compute the attention weight for the i-th state in input sequence based on key $\mathbf{K}_j$, which is denoted by:

$$\alpha_{ij} = \text{softmax}\left(\frac{\mathbf{Q}_i^\top \cdot \mathbf{K}_j}{\sqrt{d_k}}\right) \tag{14}$$

Finally, the information from other embedded features of surrounding vehicles is extracted based on attention weight and the output of multi-head attention $\boldsymbol{r_i}$ for the embedded feature of the i-th state $\boldsymbol{h}_i^{(t)}$ is calculated as follow:

$$\boldsymbol{r}_i = \sum_{j=1}^{N*k} \alpha_j \mathbf{V}_j \tag{15}$$

In this way, we can comprehensively model the influence among vehicles at different timestamps.

Training and implementation details: The number of heads in the multi-head attention layer is 8 and the input dimension of the encoder and decoder is 384. After obtaining the outputs, we choose the Mean Square Error (MSE) loss function to calculate the loss. Then, we train the model using the Adam optimizer and its learning rate starts from 0.0001, decaying 0.5 times smaller every epoch.

VI. Performance Evaluation

In this section, we first evaluate the proposed framework based on the simulation on open vehicle trajectory datasets, focusing its performance in typical traffic scenarios. Then, we conduct a field testing using remotely controlled four-wheel robots to verify its performance with real sensor measurements.

A. Simulation Study

We conduct the simulation based on two vehicle trajectory datasets, namely the Intersection Drone (inD) Dataset [34] and Highway Drone (highD) dataset [35] for evaluating the performance proposed methods. Specifically, the inD and highD datasets record naturalist trajectories of vehicles and other road users at intersections and highways, respectively. Both datasets are obtained from a bird's eye view using drones and provide for vehicles' location, velocity, and acceleration. We simulate the error of GNSS localization as a two-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}_{1\times 2}, \Sigma_{GNSS})$ by varying the value of Σ_{GNSS} to simulate the GNSS error under different situation. The error of intervehicle distance measurements detected by high-precision ranging sensor (e.g., LiDAR) also follows the Gaussian distribution $\mathcal{N}(0, \sigma_{RANGE})$ with $\sigma_{RANGE} = 0.5$. In terms of vehicle dynamics, the measurement uncertainty of velocity and acceleration for each vehicle is modeled as two independent Gaussian distributions $\mathcal{N}(0, \sigma_{ve})$ and $\mathcal{N}(0, \sigma_{acc})$, respectively, where $\sigma_{ve} = 2$ and $\sigma_{acc} = 0.2$.

To evaluate the cooperative vehicle localization, one specific location from the highD and inD datasets is selected, using all recorded trajectories at these two locations for assessment. A total of 324 vehicles over 961 seconds and 863 vehicles over 901 seconds are recorded in intersection and highway scenarios, respectively. For trajectory prediction, these two datasets are also employed for training and testing, allocating approximately 80% of trajectory data for training and the remainder for testing. The data processing for two datasets includes the following 3 steps. First, we downsample the positions in each trajectory from 25Hz to 2.5Hz. Second, we segment the trajectories with 9-second intervals. The trajectory data and corresponding sensor measurements from the first 6 seconds are utilized as input for VFormer, while the last 3 seconds are dedicated to prediction. Third, we add Gaussian noise to the positions in the training dataset for generalization.

To quantitatively evaluate the algorithm performance, the following two metrics are adopted to evaluate the localization and trajectory prediction performance, respectively.

- *Localization error:* it is defined as the Euclidean distance between each estimated vehicle coordinate and its true coordinate.
- *Rooted Mean Square Error (RMSE):* it is defined as the average distance between predicted locations and ground truth positions:

$$RMSE = \sqrt{\frac{1}{t_n}\sum_{i=1}^{t_n}\left(x_{true}^i - x_{pred}^i\right)^2 + \left(y_{true}^i - y_{pred}^i\right)^2}$$

  where t_n is the number of testing samples.

The comparison methods for cooperative localization are listed below:

- PF [9]: a particle filter based cooperative localization method that utilizes inertial sensor measurement to model state transition of particles and leverages the GNSS measurements and V2V ranging results to update the particle's weight.
- MSMV [13]: it leverages Kalman filter as a local filter to estimate vehicle's self state and a global filter to localize the target vehicle by incorporating state estimations from local filter based on the linear combination.

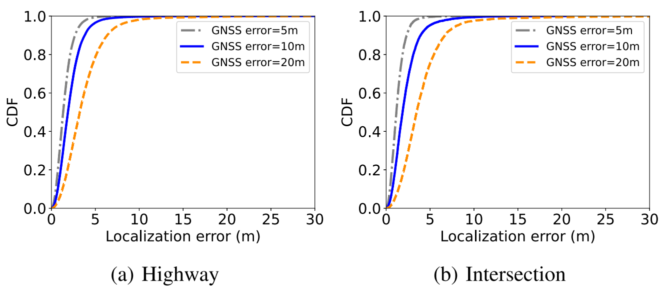


Fig. 4. CDF of localization error under different GNSS errors.

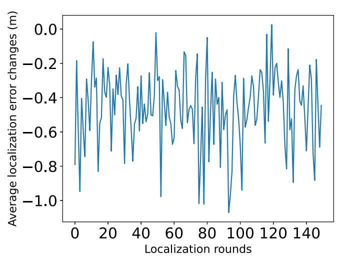


Fig. 6. localization accuracy change through backward propagation.

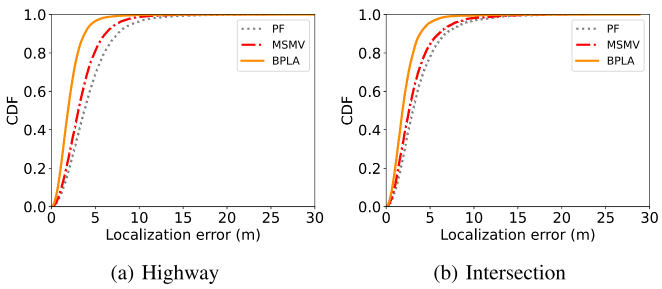


Fig. 5. CDF of localization error of different algorithms.

For vehicle trajectory prediction, we have compared the following methods:

- CS-LSTM [19]: an LSTM encoder-decoder model that uses convolutional social pooling for modelling interdependencies in vehicle motion.
- STA-LSTM [21]: an LSTM encoder-decoder model integrated with the attention mechanisms to model spatial and temporal dependencies among vehicles, respectively.
- MHA-LSTM [20]: an LSTM encoder-decoder model incorporating a multi-head attention mechanism to model the importance of neighboring vehicles and capture high order interactions among vehicles.

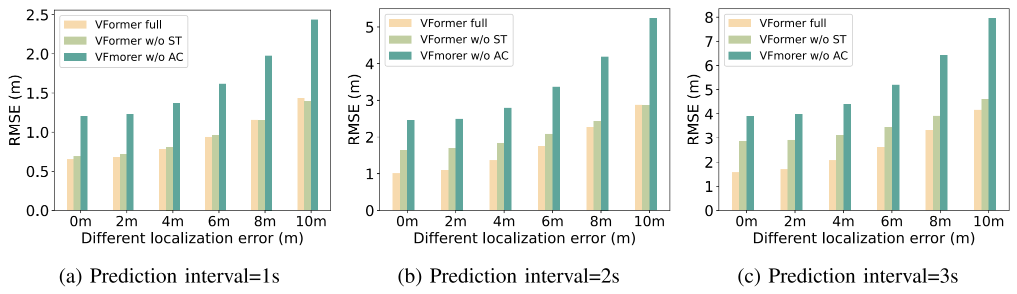
1) The Effectiveness of BPLA: We first validate the localization performance of BPLA under varying GNSS errors in two distinct scenarios. Fig. 4 shows the Cumulative Distribution Function (CDF) of BPLA localization for the two scenarios. Notably, with average GNSS errors set to 5m, 10m, and 20m, BPLA's localization error was recorded at 1.50m, 2.14m, and 3.68m for the highway scenario, and 1.31m, 2.21m, and 3.99m for the intersection scenario. BPLA has a similar localization performance at two different traffic scenarios, achieving a reduction of over 80% in localization errors compared to standalone GNSS results.

Next, we compare the localization performance of BPLA with two other cooperative localization methods, i.e., PF and MSMV. Fig. 5 shows the CDF of localization error of different algorithms when GNSS errors are set to 10m. As noted, BPLA demonstrates superior localization performance in the two scenarios, achieving an accuracy of 2.14m and 2.21m, respectively. When compared to MSMV, BPLA yields an improvement in localization accuracy of approximately 37% and 30% in the highway and intersection scenarios, respectively. Furthermore, when compared to PF, BPLA exhibits an enhancement in localization accuracy of around 50% and 40% for highway and intersection scenarios, respectively.

To assess the efficacy of backward propagation in BPLA, we record the localization errors both prior to and subsequent to the application of backward propagation, denoted by $e_{i,t}^{\text{before}}$ and $e_{i,t}^{\text{after}}$ for the i-th vehicle at t rounds, respectively. The average change in localization error at t round is computed as $e_{\text{avg}} = \frac{\sum_{i=1}^{|N_t|}(e_{i,t}^{\text{after}} - e_{i,t}^{\text{before}})}{|N_t|}$, indicating the impact of backward propagation on reducing localization errors. Fig. 6 shows the average localization error changes when GNSS error is set to 10m. As depicted, localization changes are almost all below zero, indicating that the backward propagation indeed further reduces the localization error of historical trajectories. On average, there is a decrease of 0.45m in localization error across all rounds, which represents around a 21% reduction in localization error compared to the forward phase alone.

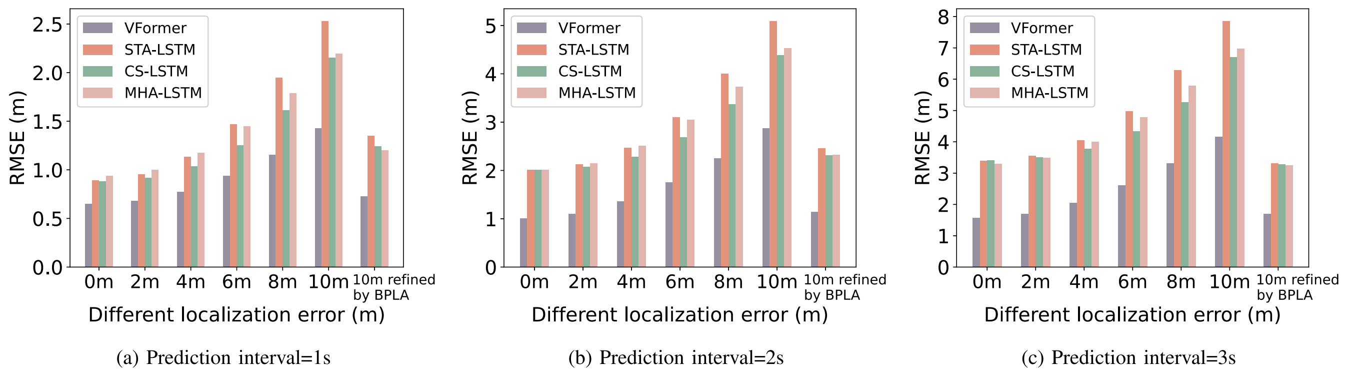
2) The Effectiveness of VFormer: In order to test the prediction performance with noisy location measurements, we add different Gaussian noise to the ground truth positions in the testing datasets for simulating different GNSS localization errors in the following experiments. We first conduct an ablation study to validate the effectiveness of the dedicated designs in the VFormer model. Specifically, we have adopted two variants of VFormer for comparison, namely, VFormer w/o AC and VFormer w/o ST. The former removes the acceleration measurements in model input and data embedding. The latter removes the spatial-temporal learning in the VFormer encoder. The prediction performance of these two variants along with the full VFormer model is shown in Fig. 7. As noted, after excluding acceleration measurements from the input, the model relies solely on noisy historical position data for predictions, leading to a notable decline in performance. On the other hand, the removal of spatial-temporal learning mechanism also decreases the prediction performance, especially in the situation that the location error is relatively low.

Next, we compare the prediction performance of different models with noisy location measurements to validate the effectiveness of VFormer. Fig. 8 and Fig. 9 shows the RMSE of four algorithms under different localization errors for two datasets. As shown, the prediction performance decreases with the increase of localization error and prediction interval. Still, VFormer achieves the best prediction performance in all scenarios of the two datasets. In highway scenarios, VFormer notably outperforms three baseline models, particularly when faced with higher localization errors. Similarly, in intersection



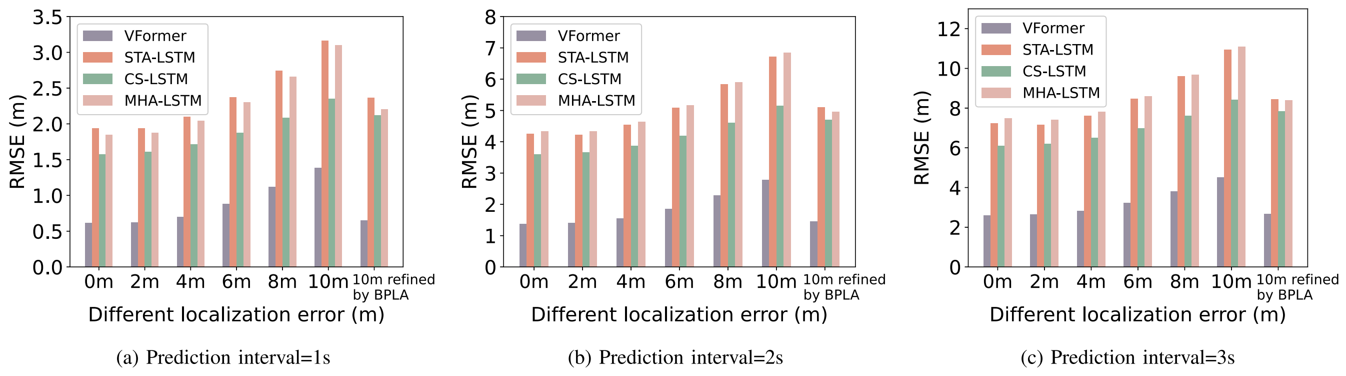
(a) Prediction interval=1s (b) Prediction interval=2s (c) Prediction interval=3s

Fig. 7. The ablation study of VFormer.



(a) Prediction interval=1s (b) Prediction interval=2s (c) Prediction interval=3s

Fig. 8. Prediction performance of different algorithms on highD dataset.



(a) Prediction interval=1s (b) Prediction interval=2s (c) Prediction interval=3s

Fig. 9. Prediction performance of different algorithms on inD dataset.

scenarios, VFormer's prediction performance is also significant superior to that of other models. These findings showcase the capability of VFormer to effectively handle noisy location measurements.

On this basis, we evaluate the overall performance of the proposed framework by using the refined localization results of BPLA for predicting the future positions. We configured the GNSS localization error at 10 meters and employed BPLA to refine GNSS localization results and construct the historical trajectories for prediction. The prediction outcomes derived from BPLA-refined trajectories are labeled as "10m refined by BPLA" in Fig. 8 and Fig. 9. As shown, the increase of localization accuracy based on BPLA can benefit all four prediction algorithms, resulting in a notable improvement of prediction performance.

Finally, we compared the training and inference times of various vehicle trajectory prediction models. The computational platform used was an Ubuntu 20.04 server equipped with an AMD Ryzen 5950x 16-core processor at 3.4GHz, an NVIDIA GeForce RTX 3090 GPU, and 64GB of memory. To emulate the computational power of RSU, we limited the RTX 3090's power consumption to 100W. Table. II displays the average training and inference times for different models on a batch of 128 samples. It is evident that the training and inference times for VFormer are considerably longer than those for LSTM-based models, attributed to its larger neural network parameters. Nonetheless, we argue that such inference time is acceptable in real-world scenarios and also meet the

TABLE II
TRAINING AND INFERENCE TIME OF DIFFERENT MODELS

	VFormer	CS-LSTM	MHA-LSTM	STA-LSTM
Training time (ms)	66.91	9.12	12.2	6.71
Inference time (ms)	22.38	1.27	1.47	1.35

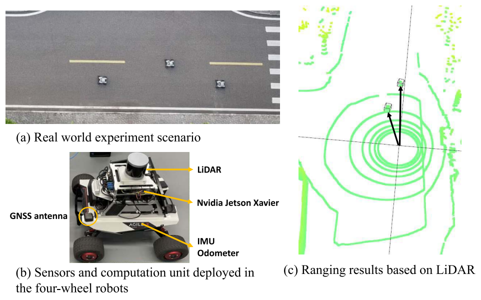

(a) Real world experiment scenario
LiDAR
Nvidia Jetson Xavier
GNSS antenna
IMU
Odometer
(b) Sensors and computation unit deployed in the four-wheel robots
(c) Ranging results based on LiDAR

Fig. 10. Field testing setup.

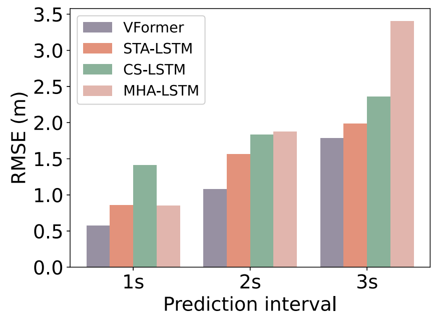


Fig. 11. Prediction performance of different algorithms in field testing.

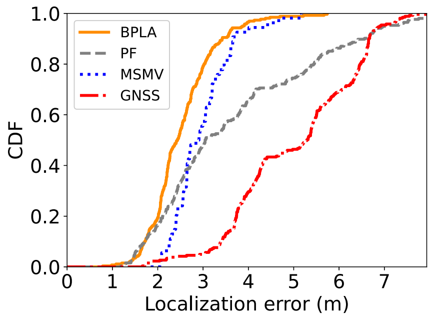


Fig. 12. CDF of localization errors of different algorithm in field testing.

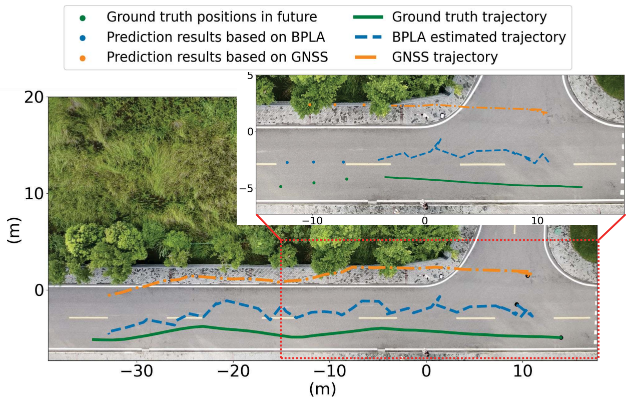


Fig. 13. Visualization of vehicle localization and trajectory prediction results.

requirements of ITS applications. Moreover, note that with an increasing of the computation capacities of RSU, the inference time overhead of VFormer could be negligible in practice.

B. Field Experiments

We have implemented a system prototype for field testing. Specifically, the target vehicles are implemented by the Agilex four-wheel robots with NVIDIA Jetson Xavier as the computing unit, as shown in Fig. 10. Sensors equipped on the wheel robots include a 16-beam LiDAR, a dedicated IMU and an odometer mounted on the chassis and a GNSS receiver. To effectively detect the Agilex wheel robots and obtain the ranging results, we train a PointPillar [36] object detection model using point cloud frames collected by the 16-beam LiDAR. The field experiment is conducted at the campus road of Chongqing University, as shown in Fig. 10. We remotely control three robots to moving on the campus road and record the sensor measurement simultaneously. Then, the collected data from different vehicles are fused to construct a dataset for evaluation. The ground truth of wheel robots' positions is captured by a drone from a bird's-eye view and is annotated manually at recorded video frames with a frame rate of 5Hz. The driving speed of wheel robots is set to around 3m/s and we have driven 3 vehicles for 5 minutes to collect the real-world sensor measurements.

We first evaluate the performance of the BPLA in the real-world scenario by comparing with PF and MSMV. The CDF of localization errors of different algorithms is shown in Fig. 12. As noted, the BPLA maintains superior localization accuracy, exhibiting average localization errors of 2.52m. In contrast, PF and MSMV yield average localization errors of 3.24m and 2.97m, respectively. The localization performance in real-world scenarios is slightly higher than the results obtained from simulations, mainly due to the significantly smaller number of cooperative vehicles during field testing. Moreover, we compare the prediction performance of different baseline models using the BPLA localization results. As depicted in Fig. 11, VFormer still achieves the best prediction performance with different prediction intervals in field testing. Finally, we visualize vehicle localization and trajectory prediction performance of the proposed framework in Fig. 13. As noted, the trajectory estimated by BPLA is much closer to the ground truth than the GNSS trajectory. Prediction results based on BPLA estimated trajectory also more closely align with the actual vehicle trajectory than prediction results based on GNSS trajectory.

VII. Conclusion

This work proposed an IoV-based cooperative localization and trajectory prediction framework for connected vehicles, which consists of a BPLA algorithm for cooperative localization and a VFormer model for trajectory prediction. The BPLA algorithm establishes a factor graph for the concern scenario and uses a modified belief propagation procedure to approximate the posterior distribution of vehicles at the current time and further refines the historical trajectories based on backward belief propagation. On this basis, the VFormer model first extracts hidden features from observed historical positions and vehicle motion data for robust prediction with noisy location measurements and then leverages multi-head attention layer to model the spatial-temporal dependencies among different vehicles at different timestamps to improve prediction performance. Both simulation study and field testing results demonstrated that the proposed framework can significantly improve localization accuracy and outperform other baseline models on trajectory prediction.

In the future work, we plan to incorporate the modeling of communication delay into the proposed framework and develop a compensation mechanism to correct errors resulting from communication delay. Furthermore, we aim to design a more cohesive framework, where prediction outcomes are incorporated as variables in the factor graph, thereby augmenting cooperative localization accuracy.

References

[1] C. Liu and K. Liu, "Toward reliable DNN-based task partitioning and offloading in vehicular edge computing," *IEEE Trans. Consum. Electron.*, early access, May 29, 2023, doi: 10.1109/TCE.2023.3280484.

[2] L. N. Balico, A. A. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. Oliveira, "Localization prediction in vehicular Ad Hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2784–2803, 4th Quart., 2018.

[3] J. Georgy, A. Noureldin, and C. Goodall, "Vehicle navigator using a mixture particle filter for inertial sensors/odometer/map data/GPS integration," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 544–552, May 2012.

[4] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 329–343, Mar. 2012.

[5] 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 Sensors J.*, vol. 22, no. 7, pp. 6818–6831, Apr. 2022.

[6] S. Bauer, Y. Alkhorshid, and G. Wanielik, "Using high-definition maps for precise urban vehicle localization," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC'16)*, 2016, pp. 492–497.

[7] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular Ad Hoc networks: A new challenge for localization-based systems," *Comput. Commun.*, vol. 31, no. 12, pp. 2838–2849, 2008.

[8] J. W. Allen and D. M. Bevly, "Performance evaluation of range information provided by dedicated short-range communication (DSRC) radios," in *Proc. 23rd Int. Tech. Meet. Satell. Div. Inst. Navig. (ION GNSS'10)*, 2010, pp. 1631–1635.

[9] S. B. Cruz, T. E. Abrudan, Z. Xiao, N. Trigoni, and J. Barros, "Neighbor-aided localization in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2693–2702, Oct. 2017.

[10] J. Liu, B.-G. Cai, and J. Wang, "Cooperative localization of connected vehicles: Integrating GNSS with DSRC using a robust cubature Kalman filter," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2111–2125, Aug. 2017.

[11] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs, and H. Wymeersch, "Implicit cooperative positioning in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3964–3980, Dec. 2018.

[12] S. Fang, H. Li, and M. Yang, "LiDAR SLAM based multivehicle cooperative localization using iterated split CIF," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21137–21147, Nov. 2022.

[13] P. Yang, D. Duan, C. Chen, X. Cheng, and L. Yang, "Multi-sensor multi-vehicle (MSMV) localization and mobility tracking for autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14355–14364, Dec. 2020.

[14] H. Kim, S. H. Lee, and S. Kim, "Cooperative localization with constraint satisfaction problem in 5G vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3180–3189, Apr. 2022.

[15] K. Liu, C. Liu, G. Yan, V. C. S. Lee, and J. Cao, "Accelerating DNN inference with reliability guarantee in vehicular edge computing," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 3238–3253, Dec. 2023.

[16] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1584–1594, Apr. 2020.

[17] T. Zhang, W. Song, M. Fu, Y. Yang, and M. Wang, "Vehicle motion prediction at intersections based on the turning intention and prior trajectories model," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 10, pp. 1657–1666, Oct. 2021.

[18] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17654–17665, Oct. 2022.

[19] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit. (CVPR'18) workshops*, 2018, pp. 1468–1476.

[20] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 175–185, Mar. 2021.

[21] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 197–208, Mar./Apr. 2021.

[22] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[23] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. C. S. Lee, "Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2544–2556, Jul. 2014.

[24] M. Brambilla, M. Nicoli, G. Soatti, and F. Deflorio, "Augmenting vehicle localization by cooperative sensing of the driving environment: Insight on data association in urban traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1646–1663, Apr. 2020.

[25] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4363–4369.

[26] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5999–6008, Jul. 2018.

[27] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020.

[28] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 236–248, Jan. 2022.

[29] Y. Li et al., "A nationwide study on cellular reliability: Measurement, analysis, and enhancements," in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 597–609.

[30] Z. Li, Y. Dai, G. Chen, and Y. Liu, *Content Distribution for Mobile Internet: A Cloud-Based Approach*. Singapore: Springer, 2016,

[31] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[32] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT, 2009.

[33] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[34] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1929–1934.

[35] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2118–2125.

[36] 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.

Feiyu Jin received the M.S. degree in computer science from Chongqing University, Chongqing, China, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include pervasive computing, mobile computing, and intelligent transportation system.

Kai Liu (Senior Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong in 2011. From December 2010 to May 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, USA. From 2011 to 2014, he was a Postdoctoral Fellow with the Singapore Nanyang Technological University, City University of Hong Kong, and Hong Kong Baptist University. He is currently a Professor with the College of Computer Science, Chongqing University, China. His research interests include Internet of Vehicles, mobile computing, and pervasive computing.

Chunhui Liu received the B.S. degree in computer science from Chongqing University, Chongqing, China, in 2019, where he is currently pursuing the Ph.D. degree with the College of Computer Science. His research interests include Internet of Vehicles, mobile edge computing, and edge intelligence.

Tongtong Cheng received the M.S. degree from Northwest Normal University, Gansu, China, in 2023, and is currently pursuing the Ph.D. degree with the Computer Science Department, Chongqing University. His research interests include Internet of Vehicles, cooperative sensing, and intelligent transportation system.

Hao Zhang (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Chongqing University, Chongqing, China, in 2015 and 2020, respectively. He is currently an Assistant Professor with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing. His research interests include pervasive computing, machine learning, and wireless sensing.

Victor C. S. Lee (Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong in 1997. He is currently a Lecturer with the Department of Electrical and Electronic Engineering, University of Hong Kong (HKU). Before joining HKU, he was an Assistant Professor with the Department of Computer Science, City University of Hong Kong. He was a Research Fellow with CityU in 1997, a Visiting Research Associate with the University of Virginia in 1999, and a Visiting Scholar with the Department of Computer Science and Information Engineering, National Taiwan University in 2008. His research interests include intelligent transportation systems, vehicular networks, e-learning, machine learning, and real-time databases. He was the Chairman of the IEEE, Hong Kong Section, Computer Chapter from 2006 to 2007. He is a member of ACM and IEEE Computer Society.