Segment-Based Trajectory Prediction and Risk Assessment for RSU-assisted CAVs at Signalized Intersections

Yue Cao , Wei Shangguan, Arnoud Visser, Junjie Chen, Linguo Chai, Baigen Cai

Preprint version, official version can be found at doi: 10.1109/TIV.2024.3414198

***Abstract*—Detecting surrounding situations and reacting accordingly to avoid collisions remains a challenging task for autonomous driving. This task requires predicting the trajectories of surrounding agents and assessing the potential risk of future situations, which can be difficult to achieve solely through onboard vehicle devices. Therefore, this paper proposes a cooperative architecture for trajectory prediction and risk assessment conducted on roadside devices (RSUs) to assist Connected and Autonomous Vehicles (CAVs). Firstly, we develop a segment-based prediction model (SegNet) tailored to hub signalized intersections. Intersections are divided into multiple segments, and the Curvilinear coordinates are utilized to indicate the geometric road features. The model leverages individual interaction cues in the ego segment and group features in the merging segments, while also incorporating traffic signal information to generate multimodal prediction results. In terms of risk assessment, we utilize the prediction results to provide hierarchical assistance, such as risk values, risk maps, and reference trajectories. Offline experimental results demonstrate that our SegNet model achieves competitive and well-balanced performance compared to state-of-the-art methods on the CitySim Database, with more accurate and smooth prediction trajectories. Through real-time CARLA and SUMO co-simulation, the performance of assisted CAVs indicates that they can safely and effectively navigate with the support of the proposed architecture.**

***Index Terms*—Autonomous Driving, Trajectory prediction, Risk assessment, Vehicle-road collaborative assistance**

I. Introduction

AUTONOMOUS driving has rapidly evolved as an effective approach to improve travel safety and efficiency. The advancements in perception, planning, and control technologies demonstrate its feasibility to improve traffic safety, relieve congestion, and reduce energy consumption in the near future [1]. However, autonomous vehicles are still striving with the challenges of maintaining safe operation in complex environments [2]. A key aspect involves predicting future trajectories of surrounding agents like pedestrians, vehicles, and bicycles to avoid potential risks. This necessitates vehicles to precisely perceive the environment, with different sensors actively combined to overcome diverse weather, lighting, and occlusion conditions [3].

In recent years, competitions and challenges like nuScenes, Argoverse [4], and Waymo have driven the development of prediction technologies [5]. However, traffic rules are rarely considered as explicit inputs to the model. Deeper extraction of traffic geometric constraints and intricate vehicle interactions is required to enhance algorithm robustness. The prediction results lack effective integration with other autonomous driving modules such as planning and control, along with insufficient details on deployment. Additionally, complex predictions impose a substantial computational burden, raising concerns about real-time performance. These limitations [6] [7], which solely rely on onboard vehicle devices, further lead to a delayed response to imminent risks [8].

Thus, the Cooperative Vehicle Infrastructure System (CVIS) leverages the prior information and powerful computing capabilities of roadside units (RSUs) to provide assistance with connected autonomous vehicles (CAVs) [3], [9]. Compared to single-vehicle autonomous driving, CVIS integrates multiple sources of information to obtain richer features, enabling more efficient and intelligent traffic management functionalities. Through rapid analysis of traffic conditions from macro and micro perspectives, RSUs can identify risks within the traffic environment, provide warnings to vehicles, and assume increasing responsibility for various modules of autonomous driving [10], [11].

At present, CVIS has been validated to reduce congestion and traffic accidents, particularly at signalized intersections, which serve as critical traffic hubs in urban areas due to their capability to control and optimize traffic flow [12]. Vehicles at intersections are regulated by traffic lights, but the mixed scenario involves more factors to consider [13]. Consequently, such attributes make them highly prospective for applications and potentially become the pioneering deployment scenarios for CVIS [14] [15].

A. Trajectory Prediction

The essence of trajectory prediction lies in maximizing the utilization of all available environmental cues to construct an accurate distribution model of future states [16]. Target agent cues, complemented by static and dynamic contextual cues, primarily constitute the inputs for trajectory prediction.

This work was supported by Equipment Preresearch Joint Fundation of Ministry of Education 8091B022238, National Natural Science Foundation of China 52272328, Natural Science Foundation of Beijing, China 23L10102 and National Natural Science Foundation of China 52102396. *(Corresponding author: Wei ShangGuan.)*

Yue Cao, Junjie Chen and Linguo Chai are with the School of Automation and Intelligence, Beijing Jiaotong University, Beijing, 100044 China (e-mail: 20111072@bjtu.edu.cn; jjchen1@bjtu.edu.cn; lgchai@bjtu.edu.cn).

Wei ShangGuan and Baigen Cai are with the School of Automation and Intelligence and State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, 100044 China (e-mail: wshg@bjtu.edu.cn; bgcai@bjtu.edu.cn).

Arnoud Visser is with the Intelligent Robotics and Computer Vision Lab of the Informatics Institute, Faculty of Science, University of Amsterdam 1098XH, the Netherlands. (e-mail: A.Visser@uva.nl).

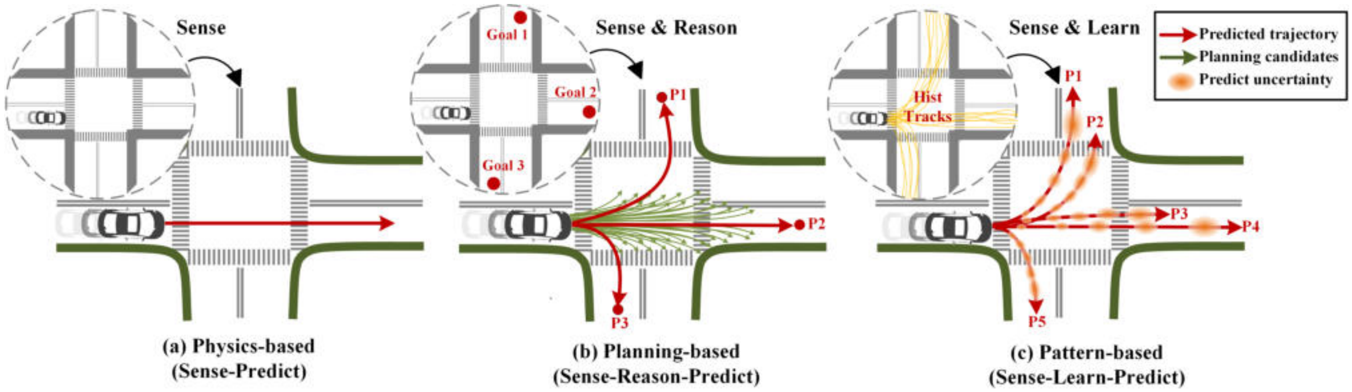


Fig. 1. Illustration of three different trajectory prediction methods.

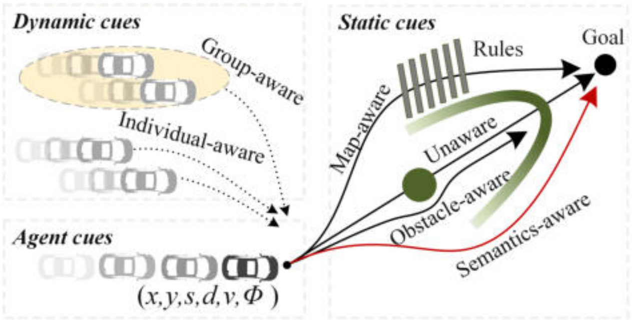


Fig. 2. Environmental cues used in trajectory planning.

Target agent cues, including position, velocity, orientation and driver-related information, are widely regarded as the most valuable features. The use of static cues varies across the literature. For unstructured road driving, obstacle data predominates as key static information. Conversely, road geometry and map information are employed to tackle the challenges associated with structured road traffic that encompasses intricate traffic constraints. Semantic cues are also incorporated in urban scenarios, such as traffic signal phases, speed limits, and restricted areas. Using dynamic cues still remains a challenge in trajectory prediction. The inherent interaction between traffic agents is highly complex, especially at intersections [17]. Different literature adopts entirely distinct strategies for handling dynamic agent information. While some research attempts to account for the impact of all nearby vehicles within a certain range [18], others take into account the social interactions among contextual vehicles for analysis [19].

Therefore, as illustrated in Fig.1, trajectory prediction methods can primarily be categorized into three approaches: physics-based methods, planning-based methods, and pattern-based methods [16], based on the strategy of utilizing different types of cues shown in Fig.2.

1) Physics-based methods: Relying on the principles of physics and mechanics, physics-based methods can be categorized into deterministic and stochastic approaches. Deterministic methods employ vehicle dynamics and kinematics models, such as ”bicycle” models and constant velocity models, to describe the motion of objects [20]. Non-deterministic employ probabilistic models, such as Kalman filters and particle filters, to account for the uncertainty in object motion. However, due to the challenge of considering complex cues, these methods commonly provide short-term estimates within 1 second [21].

2) Planning-based approaches: Planning-based methods first reason about the likely target positions, followed by the planning methods to find optimal solutions [22]. The most common methods include search-based, sample-based, and optimization-based methods. Graph search-based methods discretize the configuration space and search for an optimal solution using Dijkstra, A*, and their variants [23]. The sampling-based planning method utilizes sampling to explore the state space and find a feasible path, such as Rapidly-exploring Random Tree (RRT) [24] and parameterized curve methods [25]. Optimization-based methods generate an optimal trajectory by optimizing a certain objective function. However, planning-based methods are computationally expensive and unstable, making them difficult to apply in dynamic and complex trajectory predictions encountered at intersections.

3) Pattern-based methods: Pattern-based methods employ artificial intelligence (AI) to learn the motion characteristics of vehicles based on historical data to address these limitations. Among them, cluster-based methods aim to learn various motion trajectory patterns directly from vehicle tracks [17]. These methods utilize clustering algorithms to divide the trajectories into different clusters, each representing a group of trajectories with similar motion characteristics. During real-time prediction, matching methods are employed to identify which best aligns with the current state of a vehicle. However, these approaches require extensive data processing and struggle to consider real-time interactions among agents.

Sequential models excel in handling data with temporal order or sequential structure, leveraging memory mechanisms to capture contextual information and long-term dependencies within the sequences. Long-term trajectory prediction can be effectively modeled as a high-order Markov process, where the system’s future state depends not only on its current state but also on previous states. Therefore, sequential models are well-suited for prediction tasks. Recurrent Neural Networks (RNNs) have demonstrated remarkable proficiency in processing time series data, yielding promising outcomes in the prediction task within real-world contexts.

Florent et al. [26] were among the first to apply Long Short-Term Memory (LSTM) for trajectory prediction. Alahi et al. [27] introduced the Social LSTM model (S-LSTM) that integrates social interaction forces among pedestrians into its unimodal framework. Leveraging similarities between vehicle and pedestrian trajectory planning, Nachiket Deo [19] introduced the Convolutional Social Pooling model (CS-LSTM), designed for highway scenarios with six defined vehicle maneuvers according to their lateral and longitudinal positions. The Multiple Futures Prediction model (MFP) [18] improves the approach by learning meaningful latent variables to predict various possible futures without explicit labels, using a dynamic attention-based encoder. Planning-informed Prediction model (PiP) [28] is proposed for predicting future trajectories in a planning-informed approach, combining history tracks and the future planning sequences to perform predictions of surrounding agents. These approaches are further extended to predict trajectories at intersections based on lanes or intentions [17] [29], and also address predictions in complex urban scenarios with multiple agents [30]. However, traffic rules like traffic lights are rarely taken into account.

Furthermore, the integration of the attention mechanism dynamically learns and allocates weights based on contextual features, focusing attention on important information [31]. Multi-head Attention Social Pooling (MHA-LSTM) [32] uses multi-head dot product attention method for modeling vehicle interactions on highways. Due to their powerful capabilities in processing image and video data, Convolutional Neural Networks (CNNs) have achieved considerable success in the field of computer vision. Therefore, CNNs are frequently utilized to extract geometric features from bird's-eye view images [18], [33], [34]. Additionally, in recent years, other deep learning methods such as Graph Neural Networks (GNN) and Generative Adversarial Networks (GAN) have also demonstrated effectiveness in trajectory prediction tasks, as detailed in [6].

Notably, to ensure safety, most trajectory prediction methods are validated using databases such as NIGSIM, Argoverse, and Interaction Dataset. Simulations are sometimes used for validation [18], but those scenarios are simple with basic vehicle motion models, limiting the analysis of complex interactions under various congestion conditions.

B. Risk Assessment

Once future predicted trajectories are obtained, they can be used for vehicle risk assessment [35] [36]. In risk assessment, there are two main categories of methods: deterministic and probabilistic methods.

In deterministic methods, a binary indicator is to determine whether a collision will occur between two vehicles. The most common approach is to calculate the distance difference between discretized trajectory points at each time step, taking the vehicle's shape into account. This binary indicator is often used for candidate trajectory screening. In addition, many indicators are used to quantify potential risks, such as Time-to-Collision (TTC), Time-to-Brake (TTB), and Time-to-Steer (TTS) [13]. However, these indicators usually assume that the vehicle travels according to the physics-based models, which can result in errors in long-term prediction.

Probabilistic methods consider vehicle uncertainty by assuming a probability distribution, primarily Gaussian distribution [37] and Monte Carlo methods. Schreier et al. [38] introduced the Time-To-Critical-Collision-Probability (TTCCP) metric, a novel approach that extends TTC for uncertain multi-object scenarios with extended prediction horizons, taking into account the uncertain outcomes of all vehicular maneuvers.

Risk maps discretize the driving space in different dimensions and integrate various types of future dynamic risks from different sources. In this way, vehicles can intuitively evaluate the feasibility and superiority of trajectory candidates with different dimensional risk maps [39]. [40] build a risk field model and test in intersection Car-Following scenario. In [41], a twisted Gaussian risk model is proposed using both longitudinal and lateral motion states for vehicle behavior description. Both occupancy and flow are predicted in a spatio-temporal grid using a deep learning network [42]. However, current risk map generation methods rarely consider the prediction multimodality. [21] and [25] explored the utilization of multimodal probabilities to create risk maps, focusing exclusively on in-lane maneuvers without extending their predictions to cover intersection-related risks.

C. Contributions

Motivated by the above problems, we leverage the advantages of RSU to provide assistance to CAVs, aiming to mitigate and avoid collision risks. Integrating prior information and real-time traffic situation updates, RSUs predict future trajectories of traffic participants. Subsequently, hierarchical risk-assistive information is generated to support CAVs. The main contributions of this paper can be summarized as follows:

(1) We proposed a segment-based trajectory prediction model SegNet for signalized intersection. Employing a segmentation approach, signalized intersections are divided into multiple segments, and a Curvilinear coordinate system is used to present road geometric information. Incorporating inputs such as traffic lights, vehicle kinematics, and both individual- and group-aware interactions enables multimodal prediction of all vehicles within intersection zones.

(2) We propose a risk assessment-oriented cooperative motion planning architecture based on RSUs. Three layers of risk-assistive information, corresponding to various vehicle motion planning stages, are derived from predicted results. This process involves calculating a risk value, constructing a risk map, and formulating a reference trajectory for CAVs.

(3) We validated the methods by combining an offline dataset and real-time simulations. Offline data testing and validation are conducted using the CitySim database. Furthermore, we innovatively conduct testing using a CARLA-SUMO co-simulation and analyze three typical intersection scenarios.

The remainder of the paper is structured as follows. Section II presents the cooperative motion planning architecture and the coordinate conversion. Section III constructs the SegNet model. Section IV measures the risks. Then, Section V analyzes the test results in offline dataset and real-time simulation. Finally, the conclusions and future work are discussed in Section VI.

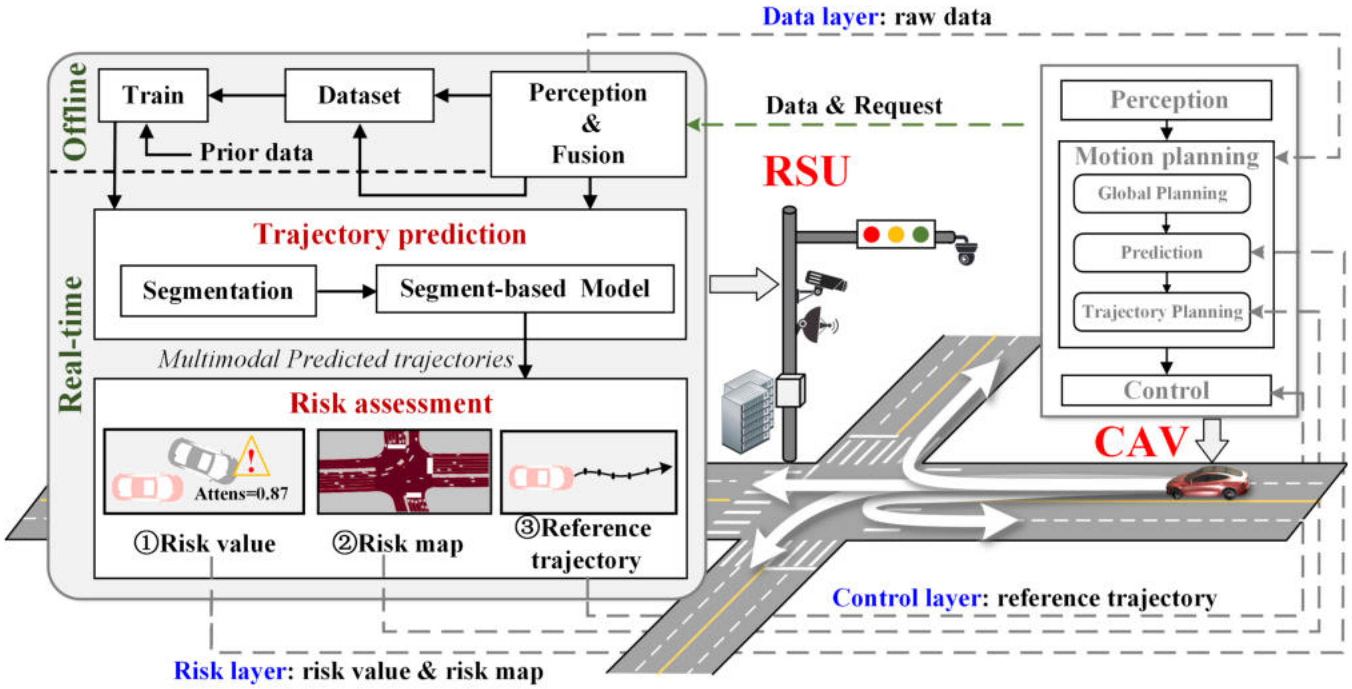


Fig. 3. The framework of cooperative motion planning. Left: The gray area illustrates the trajectory prediction and risk assessment process conducted by RSUs, which is the focus of this study. Right: various types of auxiliary information support the motion planning module of CAVs.

II. SYSTEM OVERVIEW

A. System Structure

The overall cooperative planning framework is illustrated in Fig. 3. It should be noted that in this paper, the RSUs refer to digital transportation infrastructure equipped with functions of perception, communication, computation, control, and service [43]. By enabling communication and information sharing between RSUs and CAVs, it becomes feasible to construct a bird's-eye view of the intersection. This comprehensive perspective can be achieved by integrating data from multiple roadside sensors [44] or vehicle-mounted cameras [45], employing techniques outlined in [46].

For CAVs, motion planning typically comprises global planning, prediction, and trajectory planning (or local planning) modules [47]. Based on the degree of involvement that RSU plays in the motion planning of CAV, three types are primarily divided:

Data layer: RSUs just provide prior traffic raw data, and do not participate in the planning process.

Risk layer: RSUs cooperate with motion planning but do not have a decisive role. They measure the traffic future evolution, providing collision warnings, blind spot warnings, and other information, thereby altering the attention of CAVs at the motion planning level. Additionally, computing risk maps can serve as constraints in trajectory screening.

Control layer: RSUs take over control of the CAV. RSUs can directly provide processed reference driving trajectories, while the CAV needs to judge, match, integrate, and execute these provided trajectories.

During the offline period, RSUs collect historical traffic data, encompassing road geometry, traffic rules, and signal phase change patterns. Traffic agent trajectories are processed and compiled into a dataset used for training the trajectory prediction SegNet model.

In real-time applications, CAVs send various assistance requests along with information such as the license plate number and locations for RSU matching. RSUs then deploy segmentation modules to classify CAVs. With SegNets, multimodal future trajectories of all vehicles can be obtained. Using this information, the risk assessment module computes different auxiliary data, which are then relayed back to assist CAVs. This paper primarily focuses on computing risk values and risk maps for the Risk layer, and also introduces reference trajectories for the Control layer.

B. Coordinate Conversion

On-road urban driving is highly confined to lane-divided structured roads. Therefore, the Curvilinear coordinate system (also referred to as the Frenet coordinate system), which takes the road centerline and tangential axis as the axes, is proposed to deal with curved roads [20] [25]. Fig.4 illustrates the process of converting between the Cartesian coordinate system and the Curvilinear coordinate system. For moment t_0, the conversion $\overrightarrow{p}\left(x_{ego}, y_{ego}\right) \Rightarrow \overrightarrow{p}\left(s\left(t_0\right), d\left(t_0\right)\right)$ can be represented by the following formula:

$$\overrightarrow{p}\left(s\left(t_0\right), d\left(t_0\right)\right)=s\left(t_0\right) \overrightarrow{t_r}\left(s\left(t_0\right)\right)+d\left(t_0\right) \overrightarrow{n_r}\left(s\left(t_0\right)\right) \quad (1)$$

Where the vector pairs $[\overrightarrow{n_r}, \overrightarrow{t_r}]$ are the tangential and normal vectors of the closest point. [48] extensively details the process for efficient conversion between the two coordinate systems using Newton's descent method. Utilizing both coordinate

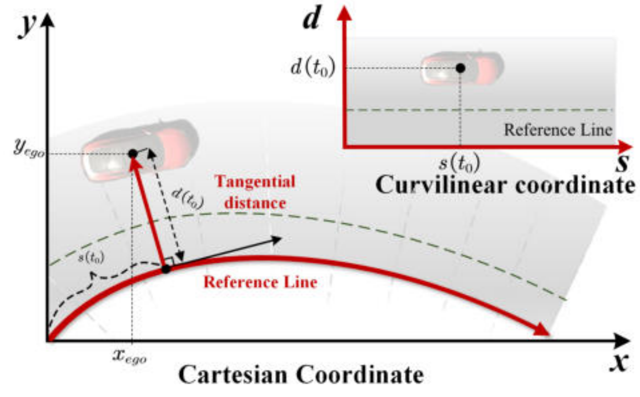


Fig. 4. Conversion between Cartesian coordinate system and Curvilinear coordinate system

systems based on data features can significantly enhance data processing efficiency. It is crucial to note that, for the purpose of distinguishing between the two coordinate systems, this paper represents positions in the **Cartesian coordinate system** with $\mathbf{x}$ and $\mathbf{y}$, while **Curvilinear coordinate system** with $\mathbf{s}$ and $\mathbf{d}$.

III. Trajectory Prediction of RSU

A. Problem Definition

The state of all vehicles n at time t within the perception range of the RSU can be represented by $I_t = \{I_t^1, I_t^2, ..., I_t^n\}$, where $I_t^n = \{x_t^n, y_t^n\}$ denotes the current state vector of the vehicle. Therefore, all historical trajectories within τ time units prior to t can be represented as $\mathbf{I_t} = \{I_{t-\tau}, I_{t-\tau+1}, ..., I_t\}$. $\mathbf{Q_t}$ represents all contextual cues at time t. The future states at predicted time T can be denoted by $\mathbf{O_t} = \{O_{t+1}, O_{t+2}, ..., O_{t+T}\}$, where $O_t = \{O_t^1, O_t^2, ..., O_t^n\}$ represents the state of all vehicles.

Thus, the essence of trajectory forecasting lies in precisely modeling $p(\mathbf{O_t}|\mathbf{I_t}, \mathbf{Q_t})$. To leverage geometric cues and incorporate multimodality, we use G representing feasible travel segments and M representing maneuvers. Hence, our proposed model aims to find:

$$p(\mathbf{O}|\mathbf{I}, \mathbf{Q}) = \sum_M \sum_G p(\mathbf{O}|M, G, \mathbf{I}, \mathbf{Q}) p(M|G, \mathbf{I}, \mathbf{Q}) p(G|\mathbf{I}, \mathbf{Q}) \tag{2}$$

For each known maneuver and segment, we employ a bivariate Gaussian distribution for modeling the future distribution, denoted as $\Theta_t = \{\mu_x^t, \mu_y^t, \sigma_x^t, \sigma_y^t, \rho^t\}$, where μ_x and μ_y are mean vectors, σ_x, σ_y are the standard deviations, and ρ is the correlation coefficient. We use the mean values μ as the predicted positions in our model. To distinguish, the object that needs to be predicted is denoted as the Ego Vehicle (EV), while other vehicles present in the environment are referred to as Surrounding Vehicles (SVs).

B. Segmentation module

In contrast to highway environments, EVs at intersections are subject to influences from SVs originating not only from the ego segment but also from other segments, exhibiting distinct characteristics. This phenomenon can be described as the "intrusion" of vehicles from other segments into the primary flow, akin to the merging process on highway ramps. Thus, segments in which the ego vehicle is located are referred to as ego segments, while segments that have a significant impact on the ego segment are referred to as merging segments.

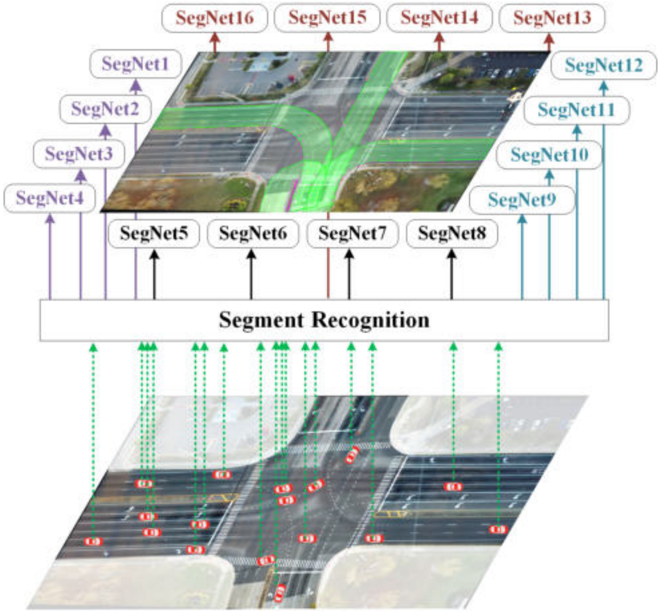


Fig. 5. Illustrate of Intersection Segmentation.

Additionally, RSUs are commonly deployed in a distributed manner, enabling segmentation to efficiently assign prediction tasks to different computational centers. On account of these considerations, our approach incorporates a segmentation model that partitions an intersection into 4x4 segments, covering 4 directions (east, south, west, north) and 4 types of feasible maneuvers (U-turn, left-turn, proceeding straight, right-turn), shown in Fig.5. For intersections that are ideally centrosymmetric, a simpler division into four segments might suffice. However, such symmetry is often too idealistic, commonly encountered in simulation environments rather than real-world applications, and hence was not adopted in our project. Meanwhile, this segmentation approach is applicable to various traffic scenarios with guiding functions, such as T-junctions and roundabouts.

Initially, we extract the road geometry and discretize it into a set of road points. Areas of higher curvature necessitate denser sampling to maintain continuity. These road points are then integrated with cubic splines to form the reference line for each segment. Based on the hist tracks in Cartesian coordinate system, the vehicles are allocated to different segments by a single LSTM encoder followed by a softmax layer. Notably, a vehicle may be assigned to multiple segments, as long as the segment probability exceeds a certain threshold (as 0.03 in this paper). Through this module, vehicle intentions at intersections are better represented.

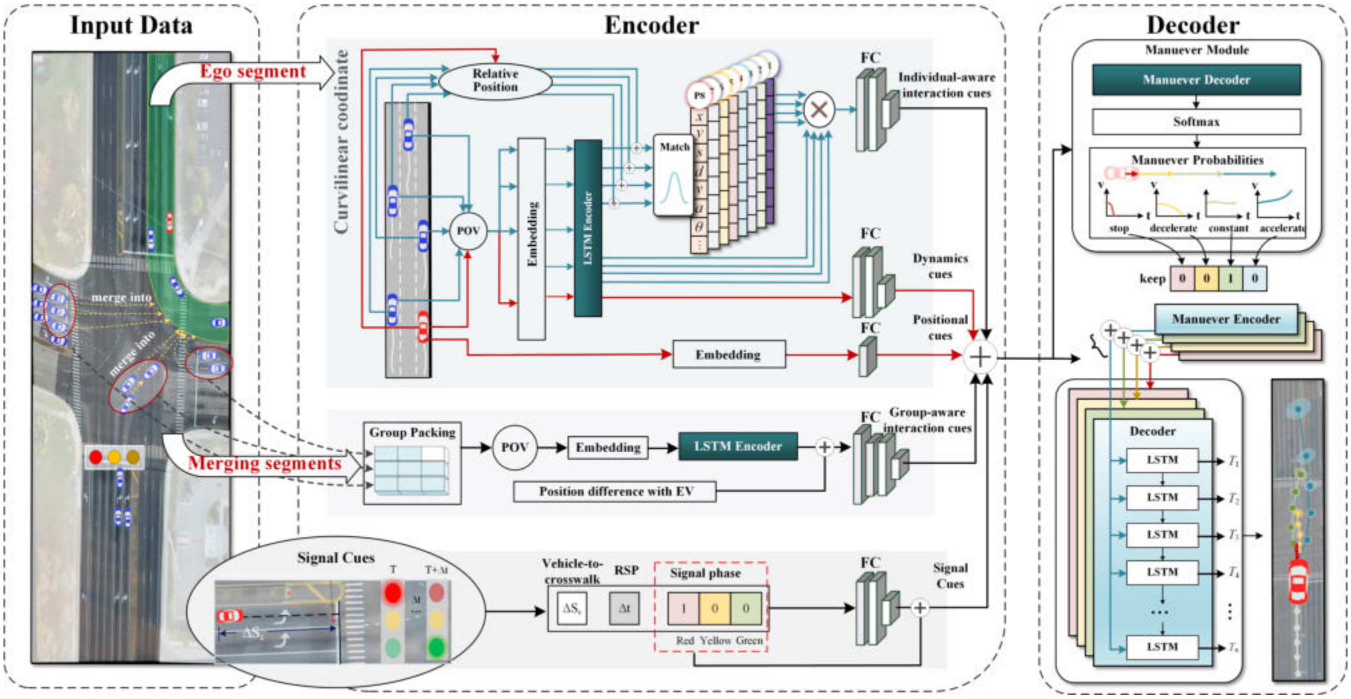


Fig. 6. Structure of SegNet.

C. Overall SegNet

Historical trajectories of all vehicles along with contextual information are collected for input, and a segmentation module categorizes the vehicles accordingly. Our model utilizes an encoder-decoder architecture, with the overall framework depicted in Fig.6.

Employing the Curvilinear coordinate system, we process the dynamic features of agent cues in the ego segment. An attention-like mechanism is used to derive individual-aware interaction features based on relative positions. Using the Cartesian coordinate system, we extract group-aware interaction features for the merging segment. Contextual cues, with a focus on signal features, are handled independently. Incorporating the one-hot maneuver encoder, multimodal results are decoded.

D. Segment-based encoder

The encoder module, depicted in the middle of Fig.6, explains how the model employs encoder modules for the ego segment, merging segment, and signal features sequentially.

1) Ego segment features: Initially, within the Curvilinear coordinate framework, a Point of View (POV) transformation normalizes each vehicle track relative to its current lateral and longitudinal positions. Following this, an embedding module maps each track into a high-dimensional vector, thereby enhancing the model's capacity to understand the intricate movement features. These vectors are further processed by an LSTM to capture their temporal characteristics. An LSTM unit can be represented as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}),\\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}),\\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}),\\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}),\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t,\\
h_t &= o_t \odot \tanh(c_t).
\end{aligned}
\tag{3}
$$

Here, i_t, f_t, g_t, and o_t represent input, forget, cell, and output gates. σ is the sigmoid function. h_t and c_t are the hidden and cell states at time t, with x_t as input. Weights (W) and biases (b) are used for computations, and $\odot$ denotes hadamard product. Vehicles within the same coordinate system exhibit consistent dynamics models, thereby sharing the same LSTM parameters. This implies that all vectors, directed toward the same LSTM encoder in Fig.6, inherit the same parameters. The kinematic model of the EV itself proves to be highly beneficial. Thus, a dedicated Fully Connected (FC) network is employed to further capture its characteristics with LSTM output hidden states. In addition, vehicles behave differently at a signal intersection as the path progresses. To this end, our model incorporates locational context, processing the current absolute position of the ego vehicle through an embedding module, succeeded by a linear layer.

It is essential to address the individual-aware interaction between EV and each SV within the ego segment. Motivated by [18], we implement a dynamic, attention-like mechanism to delineate the features of surrounding vehicles. While their model is used to identify patterns associated with various directions at intersections, our approach takes a different path by focusing on the latent effects of relative position, velocity, and turning angles of nearby vehicles within the ego segment.

We developed 8 patterns to dynamically understand these variables, tailored to the vehicle's orientation (front, back, left, right) relative to the EV, each containing 8 parameters. To match the patterns, we utilize a two-layer FC to refine the interaction features into 8 elements. Then, these patterns are then matched against them in the following way:

$$\phi(\mathbf{x}) = e^{-(\epsilon\|\mathbf{x}-\mathbf{x}_{pattern}\|)^2} \tag{4}$$

Where ϵ is a scaling factor. Leveraging the attention-like mechanism enhances the comprehension of interaction features of SVs in the ego segment. Subsequently, a three-layer FC network further refines the feature vectors.

2) Merging segment features: Initially, SVs within merging segments are filtered by calculating their relative distances within the Cartesian coordinate system. Subsequently, these vehicles are organized into a group based on the number of merging segments and their capacity. For a typical four-way intersection, each ego segment corresponds to three merging segments. Therefore, we use a 3x3 group grid, which considers the three nearest SVs within each merging segment relative to the midpoint of the exiting crosswalk. The group inputs are transformed by POV, after which independent LSTM units are deployed to capture temporal features. By integrating features related to the relative distances to the EV, group-aware interaction features are obtained through a four-layer FC network.

3) Signal context features: One of the great benefits of utilizing RSUs is the integration with traffic signals, enabling access to information that may be difficult for vehicles to obtain directly, such as the Remaining Signal Phase Time (RSP). In this module, we first determine the vehicle-to-crosswalk distance, assigning a value of -999 if it has passed the crosswalk. Traffic light phases are represented using one-hot encoding for red, yellow, and green. We aggregate them and then process them through a three-layer FC network to extract their contextual features. Ultimately, these features, together with one-hot encoded signal phase information, constitute the module's output. Such attributes are crucial in facilitating the prediction of a vehicle's intention to decelerate, stop, or startup.

E. Multimodal Decoder

The segmentation module has already categorized the vehicle's maneuvers based on segments. We further conclude four maneuvers: *stop*, *deceleration*, *constant speed*, and *acceleration*. The Maneuver module independently extracts vehicle maneuvers and employs a linear network to reduce dimensions to 4. Subsequently, a softmax layer normalizes it to obtain the probability $p(M|G, \mathbf{I}, \mathbf{Q})$ for each maneuver.

In the trajectory decoder, we combine the one-hot encodings of each maneuver (e.g., representing constant speed as [0,0,1,0]) with the aggregated features. The LSTM decodes each integrated input for future timesteps. Subsequently, a linear network performs dimensionality reduction on the features to generate a bivariate Gaussian distribution, denoted as $p(\mathbf{O}|M, G, \mathbf{I}, \mathbf{Q})$. This process results in a multimodal distribution, as illustrated in the lower right of Fig. 6.

F. Training

Our training is divided into two phases. Segmentation models are trained independently to measure $p(G|\mathbf{I}, \mathbf{Q})$ by minimizing the cross-entropy loss function.

$$L_{CE}^{G} = -\sum_{i=1}^{G} g_i \log(\hat{g}_i) \tag{5}$$

Where g_i is a binary indicator (1 if sample i is of segment G and 0 otherwise), and $\hat{g}_i$ is the model's predicted probability belongs to segment G.

On the other hand, for each SegNet model, we integrate the maneuver loss L_{CE}^{M} and trajectory loss L_{NLL} to formulate the comprehensive model loss function, which the networks are trained to minimize. The trajectory loss is quantified using the Negative Log Likelihood (NLL) of the vehicle's conditional distribution. Given that the maneuver for each dataset is fixed and unique, a cross-entropy loss function L_{CE}^{M} corresponding to maneuver m_i is employed.

$$\begin{aligned} L_{train} &= -log(\sum_{i=1}^{m_i} p_{\Theta}(\mathbf{O}|m_i, G, \mathbf{I}, \mathbf{Q})p(m_i|G, \mathbf{I}, \mathbf{Q})) \\ &= L_{NLL} + L_{CE}^{M} \\ L_{NLL} &= -log(p_{\Theta}(\mathbf{O}|m_{true}, g_{true}, \mathbf{I}, \mathbf{Q})) \\ L_{CE}^{M} &= -\sum_{i=1}^{M} m_i \log(\hat{m}_i) \end{aligned} \tag{6}$$

IV. Risk assessment

With SegNets, the multimodal predicted trajectories of vehicles within the perception range of RSU at the intersection are obtained. Based on these multi-dimensional information, RSUs can provide more comprehensive support for the motion planning part of the vehicle. In this paper, we introduce three types of auxiliary information: risk value, risk map, and reference trajectory, corresponding to the Risk layer and Control layer discussed in Section II.

A. Risk value

Like Advanced Driver Assistance Systems (ADAS), RSUs can provide vehicles with warnings. Acknowledging future motions enables the quantification of collision risk between the EV and SVs. This can be described as the $collision$ risk value if the EV travels as planned $state$, considering the maneuvers of SVs M:

$$C(collision|state) = \sum_{M} C(collison|state, M)p(M|state) \tag{7}$$

Through the prediction module, we obtain the future multimodal trajectories of the SVs, where the probability of each maneuver $p(M|state)$ is represented as P_m in Fig.7. As illustrated in Fig.7, the future closest encounter distance between the two vehicles $\Delta D_{t_c}^{M}$, as well as the corresponding time of encounter t_c^M, lateral distance difference $\Delta d_{t_c}^{M}$, and longitudinal distance difference $\Delta s_{t_c}^{M}$, for each maneuver can

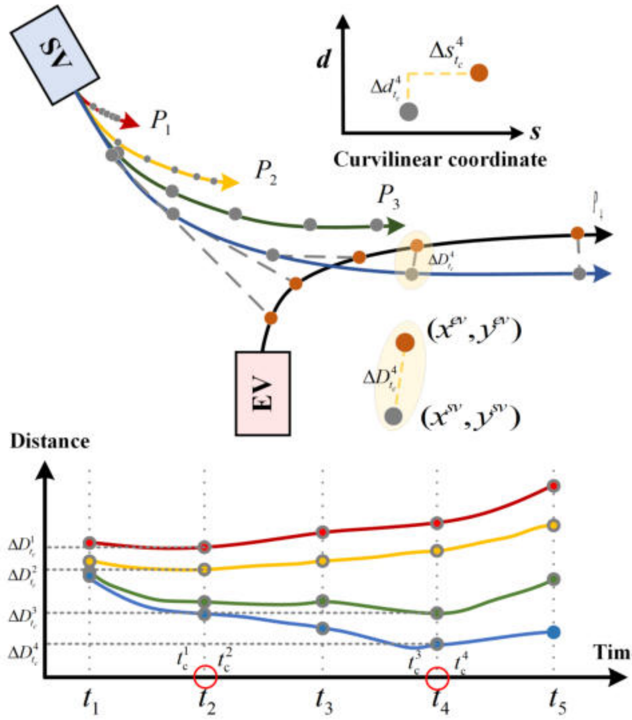


Fig. 7. Risk related parameters calculation with spatiotemporal multimodal prediction trajectories.

all be calculated. The t_c^M is dependent on the prediction step and varies for different maneuvers.

In intersections, vehicles have varying impacts on the EV across different segments. For instance, within the ego segment, considerations are akin to those on a highway, primarily focusing on the vehicle's lateral and longitudinal behavior relative to EV. However, in other segments, road guidance and orientation become important considerations. Therefore, we adopt different approaches for handling these two ways.

1) Ego segment: Vehicles in this segment are traveling in the same direction, which means that the lateral and longitudinal relationship between vehicles can be better analyzed in the Curvilinear coordinate system. Therefore, for vehicles in the same lane:

$$C^{ego} = \sum_M \frac{A_c}{\sqrt{\Delta d^2 + \left(\xi(v^{ev})\Delta s\right)^2}} p(M|state) \tag{8}$$

Where A_c is a constant number. The lateral and longitudinal distance difference are calculated by $\Delta d = \max(\left|\Delta d_{t_c}^M\right| - \frac{l^{ev}+l^{sv}}{2}, 0)$ and $\Delta s = \max(\left|\Delta s_{t_c}^M)\right| - \frac{L^{ev}+L^{sv}}{2}, 0)$. l and L represent the length and width of a vehicle. For structured on-road driving, lateral and longitudinal distances have different mapping coefficients for risk. We adopt a concept of virtual distance, which magnifies differences in the longitudinal direction to match the lateral risk:

$$\xi\left(v^{ev}\right) = \xi_0\left(v^{ev}\right) e^{-\beta(v_{ev}-v_{sv})\frac{s^{ev}-s^{sv}}{|s^{ev}-s^{sv}|}} \tag{9}$$

$$\xi_0\left(v^{ev}\right) = \begin{cases} \frac{d_0}{T_f v_{ev}} & v^{ev} \geq \frac{d_0}{T_f} \\ 1 & otherwise \end{cases} \tag{10}$$

Where β is a scaling factor, The quantity $\xi_0\left(v^{ev}\right)$ takes the absolute velocity of EV into account and adjust $\xi\left(v^{ev}\right)$ when the vehicles are traveling at the same speed. d_0 and T_f represent the vertical impact distance and desired following time, respectively.

2) Merging segments: Calculating risk in merging segments significantly differs from that in the ego lane, primarily due to the heightened need to consider orientation and road guidance. By factoring in vehicle orientation and velocity, we calculate TTC in Cartesian coordinate for evaluating vehicle risk:

$$C^{mer} = -A_s \sum_M \log(\psi * TTC_{t_c^M}) p(M|states) \tag{11}$$

where A_s and ψ jointly align the scale with ego risk. At the time t_c^M in the future, the relationship between the position and relative speed between the two vehicles needs to be considered, namely, $\vec{v} = (x_{t_c}^{sv} - x_{t_c}^{ev}, y_{t_c}^{sv} - y_{t_c}^{ev})$ and $\vec{d} = (v_{x,t_c}^{ev} - v_{x,t_c}^{sv}, v_{y,t_c}^{ev} - v_{y,t_c}^{sv})$ When $\vec{v} * \vec{d} < 0$, the two vehicles are moving away. When $\vec{v} * \vec{d} > 0$, we decompose the velocity of EV and EV along the x-axis and y-axis and get $v^{ev} = v_x^{ev}\cos\theta_{t_c} + v_y^{ev}\sin\theta_{t_c}$ and $v^{sv} = v_x^{sv}\cos\theta_{t_c} + v_y^{sv}\sin\theta_{t_c}$, Where θ_{t_c} represents the angle between the position vector and the x-axis. Then TTC can be calculated by the following formula:

$$TTC_{t_c} = \left|\frac{\Delta D_{t_c}}{(v_x^{ev} - v_x^{sv})\cos\theta_{t_c} + (v_y^{ev} - v_y^{sv})\sin\theta_{t_c}}\right| \tag{12}$$

Finally, the risk values are normalized to a range of 0 to 1 using a Sigmoid function. In practical applications, graded thresholds can be set to provide alerts and warnings to the EV, and it can serve as an indicator for assessing the safety performance of autonomous driving. Additionally, it can be used as an attention value transmitted to the EV for planning and decision modules. In this paper, this result will also serve as a parameter influencing the calculation of the risk map.

B. risk map

The spatiotemporal risk map rasterizes the environment at different resolutions and then evaluates the potential risk values for each occupied grid. RSUs can provide detailed and rapid feedback on changes in the scene offline based on pre-known information. Static risks, such as the risk of a collision with the road and the risk of staying within lane boundaries, can be calculated in advance.

$$R_{road} = \frac{A_{road}}{2} \sum_{i \in E} \left(\frac{H}{d - d_i^e}\right)^2 \tag{13}$$

Where A_{road} is weight factor. d_i^e is the lateral offset of the i^{th} edge in set E which contains the boundary of the road.

On the contrary, moving traffic agents, including vehicles, motorcycles, bicycles, and pedestrians, introduce a significant level of uncertainty into the traffic flow, making it challenging to quantify their associated risks. This paper primarily focuses on vehicles. Fig.8 illustrates the risk map calculation schematic, showing computation methods for the ego segment on the left and the merging segment on the right.

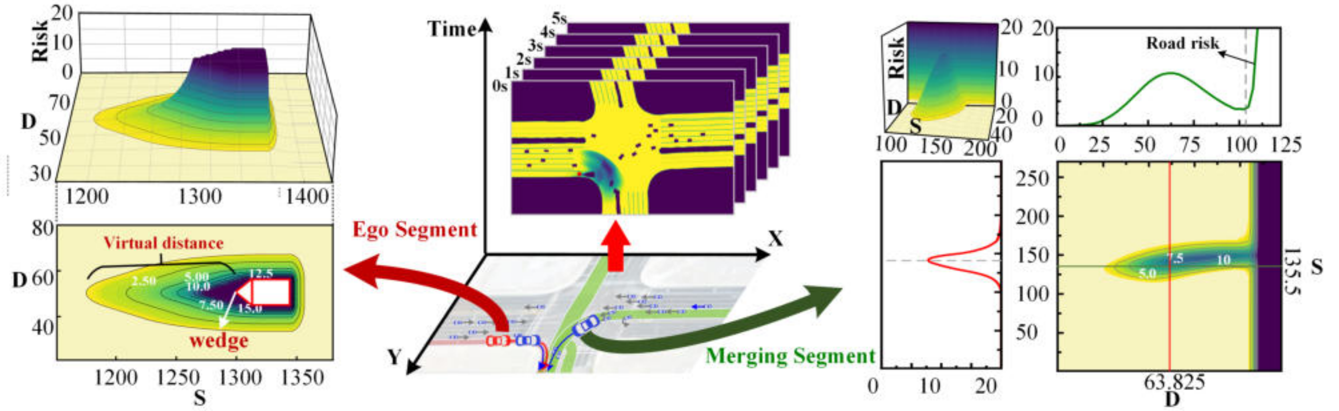


Fig. 8. Risk Map Generation: Vehicles are categorized into the ego segment and merging segment, as shown in the middle. The left side depicts gradient and 3D representations of the risk map for the ego segment, while the right side illustrates the risk map and risk curve generated in the merging segment.

1) Ego segment: For vehicles in the ego segment, where main considerations involve lateral and longitudinal maneuvers along the road, we simplify by using the concept of virtual distance to generate the risk map. The left of Fig.8 illustrates a risk map depicting the risk from a leading vehicle to the EV, where the virtual distance varies with the speeds of two vehicles. And the potential risk influence of the forward SV should not be linear and evenly distributed in the lateral direction. Therefore, a wedge can be appended to the affected side of the SV to better describe the impact of risks [49].

$$R_{car}^{ego} = A_{car}\frac{e^{-\alpha K}}{K} \tag{14}$$

$$K = \underset{(d_i,s_i)\in B}{min}\left(\sqrt{(d-d_i)^2+(\xi(v_{ev})\,s-s_i)^2}\right) \tag{15}$$

Where A_{car} is a weight factor and α is a scale. (d_i, s_i) is the position of point i in set B comprising the edge of the SV.

2) Merging Segments: For vehicles in the merging segment, we aim to calculate the risk they pose to the EV's path. Therefore, we employ the Gaussian distribution in advantage of taking uncertainty into account, which is influenced by factors such as the encounter time, minimum distance, and driving speed.

$$\begin{aligned} R_{car}^{mer} &= A_{car}\sum_{M} P(encounter|t_c^M, M, state) \\ & p(t_c^M|M, state)p(M|state)p(damage|encounter) \end{aligned} \tag{16}$$

$$p(encounter|t_c^M, M, state) = e^{-\frac{1}{2\sigma_1^2}\Delta D_{t_c}^{M\,2}} \tag{17}$$

$$p(t_c^M|M, state) = e^{-\frac{1}{2\sigma_2^2}t_c^{M\,2}} \tag{18}$$

$$p(damage|encounter) = e^{-\frac{1}{2{\sigma_3}^2}v_{ev}(t)^{-2}} \tag{19}$$

Where $p(encounter|t_c^M, M, states)$ quantifies the risk attributed to the proximity between vehicles during an encounter, with shorter distances correlating with an escalation in risk levels. $p(t_c^M|M, states)$ assesses the encounter time, with larger values suggesting longer reaction times and thus lower risk. Additionally, $p(damage|encounter)$ represents the risk to the driver due to kinetic energy in the event of a collision, with higher ego speeds $v_{ev}(t)$ resulting in greater danger. σ_1, σ_2, and σ_3 are model parameters.

On the right side of Fig.8, the risk calculation for vehicles in the merging segment is depicted. It is worth mentioning that we calculate the risk on the segment where the SV applies to the EV's route. Therefore, this risk is not necessarily aligned with the SV's direction of travel. On the contrary, the areas where the two vehicles encounter or collide will have a higher level of risk. The red and green curves in Fig.8 represent lateral and longitudinal risk curves in the Curvilinear coordinate system. Due to symmetry around the SV at turning points, the vehicle's risk can be decoupled into lateral and longitudinal curves resembling Gaussian distributions (which are actually fused by three Gaussian distributions).

By aggregating the risks associated with different vehicle segments, we can provide varying levels of risk map assistance for the trajectory planning of EVs. It is worth noting that, to account for the curvature of the vehicle's driving route, certain risks are more effectively calculated in the Curvilinear coordinate system and subsequently converted to the Cartesian coordinate system. In this paper, we perform the temporal revolution of the risk map by adopting the same step as the predicted trajectory. Additionally, we utilize the spatiotemporal risk map to restrict the candidates.

$$Pot\,(d,s,t) = R_{car}^{ego}\,(d,s,t) + R_{car}^{mer}\,(d,s,t) + R_{road} \tag{20}$$

$$Pot\,(d,s,t) \leq A_{risk} \tag{21}$$

Where $Pot\,(d,s,t)$ Represents the risk for the longditudinal and lateral positions in the Curvilinear coordinate system at future time. A_{risk} eliminates candidates with high risks.

C. Reference Trajectory

In the CVIS, the highest level of assistance that RSUs can provide is to directly take over the CAVs. This ideal strategy can maximize the capacity of traffic flow and minimize the randomness of traffic evolution. However, the trajectories directly output by the prediction module are often discontinuous in curvature, which makes it difficult to satisfy the kinematics and dynamics model of a vehicle. Methods proposed in [25] [50]

TABLE I
NLL AND RMSE COMPARISON RESULTS OF PREDICTION MODELS OVER A 5-SECOND PREDICTION HORIZON

Metric	Horizon(s)	CV	S-LSTM	V-LSTM	CS-LSTM	V-LSTM(M)	CS-LSTM(M)	CS-LSTM(M)+i	PiP(M)	MHA-LSTM(M)	MFP -1	MFP -3	MFP -5	MFP -5+i	SegNets
NLL(nats)	1 sec.	-	2.70	1.04	0.61	0.06	-0.31	-0.29	-0.04	-0.01	0.55	-0.57	-0.65	-0.56	**-0.68**
	2 sec.	-	5.54	3.36	1.98	1.12	0.24	0.23	0.58	0.59	1.92	-0.08	-0.14	-0.08	**-0.33**
	3 sec.	-	6.96	5.14	3.13	2.14	0.57	0.54	1.00	0.98	3.04	0.18	0.06	0.20	**-0.21**
	4 sec.	-	8.02	6.18	3.99	3.08	0.81	0.72	1.44	1.33	3.98	0.42	0.33	0.43	**0.01**
	5 sec.	-	9.29	6.96	4.69	3.94	1.05	0.88	1.94	1.71	4.92	0.93	0.69	0.83	**0.36**
RMSE(m)	1 sec.	6.09	2.86	0.48	0.42	0.48	0.40	0.42	0.49	0.48	0.67	**0.36**	**0.36**	0.37	0.39
	2 sec.	11.52	5.22	1.25	1.00	1.22	0.90	0.92	1.15	1.11	1.38	**0.82**	0.84	0.83	0.88
	3 sec.	16.17	7.28	2.40	1.81	2.343	1.58	1.68	2.09	2.02	2.58	1.50	1.55	1.59	**1.41**
	4 sec.	19.95	8.99	4.05	2.99	3.93	2.74	3.01	3.32	3.26	3.95	2.54	2.58	2.71	**1.97**
	5 sec.	22.70	10.28	6.05	4.57	5.93	4.27	4.59	4.81	4.77	5.72	4.19	4.30	4.33	**2.68**

can well perform trajectory optimization based on a reference trajectory. In this paper, we directly use the prediction results as the reference trajectory for vehicle control input. But if the risk constraints are not satisfied, a re-planning method will be called. The planning section is not the main focus of this paper, as detailed explanations can be found in [47].

V. EVALUATION

To apply the cooperative motion planning system, it begins with training a prediction model using collected historical traffic data. Subsequently, the trained model is deployed in real-time traffic scenarios. Therefore, we validate the performance of our proposed method through offline and real-time verification using a database and simulation, respectively.

A. Dataset Evaluation

1) CitySim Dataset: CitySim dataset is an open-source drone video trajectory and co-simulation dataset [51]. Innovatively, to the best of our knowledge, this dataset is the only one capable of providing signal timing data and CARLA&SUMO base maps in hub intersections.

Moreover, the dataset collects bird-view naturalistic driving data on more than a dozen locations in the USA and other countries. More specifically, the area of interest is the large signalized intersection in Orlando, Florida, covering an approximate area of 500x100 meters and comprising nine lanes spanning four directions. Currently, it contains vehicle trajectories extracted from over 60 minutes of drone videos recorded at 30Hz. Specifically, it captures a period from 5:40pm to 6:42pm, encompassing mild, moderate, and congested traffic conditions. Each of them includes the position of the center and the four bounding box vertex of the vehicle, as well as the speed, heading, and lane ID. The dataset is divided into 14 subdatasets, excluding two with insufficient U-turn data. Each subdataset is randomly split into 70% for training, 10% for validation, and 20% for testing purposes.

2) Implemention Details: The dataset was firstly downsampled by a factor of 6. We employ a trajectory history of 3 seconds and a prediction horizon of 5 seconds. Additionally, The prediction models are trained using Adam optimizer which the learning rate decreases exponentially with epochs. The initial learning rate is 0.001 and the minimum is 0.0003. The dimensions of the encoder and decoder LSTMs are 64 and 128, respectively. A batch size of 128 is used. The Leaky-ReLU activation is used as the activation function for all the LSTM cells. The models are trained on two RTX3080 GPUs with Pytorch implementation.

3) Evaluation Metric: We evaluate our results with two different metrics for all the predicted position points.

The root mean square error (RMSE) calculates the average difference between predicted and actual positions within a given prediction horizon t_h. We use the maneuver with the highest probability for calculating:

$$L_{rmse} = \sqrt{\frac{1}{t_h}\sum_{t=1}^{t_h}(x_{true}^t - x_{pred}^t)^2 + (y_{true}^t - y_{pred}^t)^2} \quad (22)$$

The negative log-likelihood (NLL) quantifies the degree of fit between the bivariate Gaussian distribution predictions and the real one. A lower NLL value indicates a better fit of the model to the true trajectories. It can be negative as we used a continuous density function done in [27].

4) Models Compared: we compare the performance of our proposed model with the following models:

- *Constant Velocity (CV)*: This method assumes the vehicle maintains a constant velocity. This is the simplest baseline used for comparison.
- *Vanilla LSTM (V-LSTM)* [26]: This model simply uses the previous tracks of EV in the encoder LSTM.
- *Social LSTM (S-LSTM)* [27]: employs an encoder-decoder framework that only takes into account the social interaction forces of SVs.
- *Convolutional Social Pooling LSTM (CS-LSTM)* [19]: employs social convolutional pooling to handle complex interactions in the encoder-decoder framework. This model generates multimodal predictions based on 2 longitudinal and 3 lateral maneuvers. If a model adopts multimodal output, denoted as (M).
- *Planning-informed trajectory (PiP)* [28]: This model combines a planning-informed approach by incorporating the future planning of the controllable agent.
- *Multi-head Attention Social Pooling (MHA-LSTM)* [32]: This model employs a multi-head dot product attention mechanism to highlight the surrounding vehicles. We add a multimodal output module to facilitate comparisons.

TABLE II
DETAILED NLL/RMSE RESULTS OF DIFFERENT SEGNETS

Zone	SegID	Extracted Number	Metric: NLL(nats) / RMSE(m)				
			1s	2s	3s	4s	5s
U-turn	9	5717	-0.72 / 0.51	-0.46 / 0.88	-0.26 / 1.36	-0.21 / 2.23	-0.14 / 2.48
	13	15001	-1.56 / 0.52	-1.51 / 1.04	-1.48 / 1.49	-1.43 / 2.03	-1.37 / 2.57
Left turn	2	493688	-1.23 / 0.43	-1.2 / 0.92	-1.17 / 1.43	-1.14 / 1.98	-1.09 / 2.55
	6	141973	-1.67 / 0.32	-1.66 / 0.62	-1.64 / 0.99	-1.63 / 1.23	-1.61 / 1.99
	10	208907	-1.49 / 0.35	-1.47 / 0.63	-1.45 / 0.94	-1.43 / 1.25	-1.41 / 1.69
	14	248485	-1.00 / 0.42	-0.87 / 0.86	-0.79 / 1.30	-0.73 / 1.63	-0.69 / 2.68
Keep straight	3	320638	-0.61 / 0.42	-0.6 / 0.82	-0.59 / 1.48	-0.57 / 2.12	-0.56 / 2.63
	7	278716	-1.37 / 0.39	-1.37 / 0.83	-1.36 / 1.24	-1.35 / 1.76	-1.33 / 2.38
	11	1098593	-1.00 / 0.36	-0.89 / 1.03	-0.82 / 1.50	-0.77 / 2.03	-0.72 / 2.76
	15	180997	-1.17 / 0.39	-1.14 / 0.81	-1.14 / 1.45	-1.13 / 2.07	-1.11 / 2.94
Right turn	4	11108	0.50 / 0.39	0.52 / 0.64	0.53 / 1.00	0.55 / 1.64	0.61 / 2.75
	8	21461	-0.28 / 0.25	-0.28 / 0.40	-0.26 / 0.66	-0.22 / 1.19	-0.14 / 2.24
	12	231271	-0.27 / 0.44	-0.27 / 0.76	-0.26 / 1.20	-0.22 / 1.77	-0.17 / 2.29
	16	154482	0.21 / 0.35	0.31 / 0.73	0.42 / 1.35	0.51 / 2.02	0.55 / 3.01

- *Multiple Futures Prediction with K-latent modes (MFP-K)* [18]: This model efficiently learns latent future motion modes of agents using a dynamic attention-based state encoder. In addition, map information is further utilized by applying a three-layer CNN for feature extraction. If a model incorporates map image, denoted as +i.

5) Results: Table.I shows the experimental results. Here, we can compare the performance of different methods based on the use of cues in the input, as well as the multimodal maneuvers in the output. It is worth mentioning that we attempted training and testing with the Gated Recurrent Unit (GRU) replacing LSTM, but the performance still lags behind LSTM.

In the context of self-motion information, both the CV model and S-LSTM model disregard the vehicle's tracks entirely, resulting in the two worst performances among all the models evaluated. The RMSE can even reach as high as 22.7m. These findings underscore the utmost importance of incorporating the vehicle's kinematic and dynamic characteristics into trajectory planning tasks.

In terms of leveraging road features, CS-LSTM(M)+i and MFP-5+i employ a three-layer CNN network to extract map image features. In comparison to CS-LSTM(M), CS-LSTM(M)+i exhibits improvements in terms of NLL performance, with a decrease of 16.2% for 5-second long-term predictions. But there is a 7.5% increase in RMSE, indicating map image contributes to the uncertainty of maneuver recognization. However, the incorporation of map image data does not always guarantee improved performance. In the case of MFP-5+i, this addition has actually resulted in a decrease in performance. We construct the Curvilinear coordinate system for each segment to fully leverage the geometric information of the road. We categorized the segments into four types: U-turn, left turn, go straight, and right turn. It is worth mentioning that Segment 1 and 5 were excluded from the analysis due to insufficient data (less than 2000 extracted trajectories).

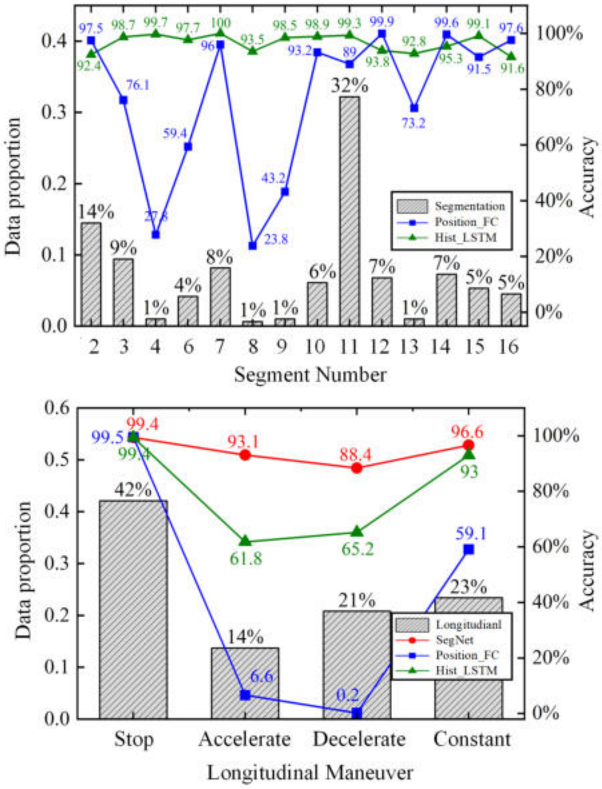


Fig. 9. The upper illustrates the segment data distribution and recognition accuracy comparison, while the lower part presents the vehicle maneuver pattern distribution and recognition accuracy comparison.

As shown in Table.II, the detailed results for each SegNet are presented according to the numbering in Fig.5. It is important to note that the NLL results are trained within their respective Curvilinear coordinate systems, thus exhibiting distinct characteristics. The RMSE in Cartesian coordinate enables comparison on the same scale. The performance of the right-turn segments is the weakest, largely due to the complex interactions involved in right turns, especially with merging segments. In addition, the performance in the right turn models are also influenced by pedestrians, which were not considered in this paper. Due to the nonlinear transformation between the two coordinate systems, we directly estimate the distribution in the new coordinate system through Monte Carlo simulation, thereby increasing the error. This superior performance is largely attributed to the fact that the SegNets fully leverage the road geometry configuration and traffic signal information. Drawn in the upper part of Fig.9, We observed that the majority of vehicles opt to go straight, representing 54.9% of the total dataset. We also compared the prediction accuracies by two ways: use a single LSTM model to abstract time-series hist features (Hist_LSTM) and a fully connected network based on the current position (Position_FC). This comparison was performed by matching the maximum predicted probability with the actual one. It can be observed that Hist_LSTM accurately classifies the segments where vehicles travel, with a minimum accuracy of 91.6%.

When considering vehicle interactions, CS-LSTM, PiP, MHA-LSTM and MFP models, in comparison to the V-LSTM method, incorporate the mutual interactions of SVs in distinct manners. The evaluation results confirm the importance of intervehicle interaction cues in accurately predicting vehicle behavior, regardless of the presence of multimodality. On the other hand, the way of handling interaction cues also plays a pivotal role in determining the predictive performance. Among the maneuver-aware models, CS-LSTM(M) employs social pooling to effectively process SV cues based on relative lateral and longitudinal distances. In contrast, the MFP model learns and matches the latent influences of SVs, taking into account their directions and positions. MHA-LSTM(M) pays increased attention to important SVs based on their contextual features. PiP(M) utilizes the interaction between planned future trajectories and surrounding vehicles. Consequently, in CitySim dataset, the MFP model outperforms the other three methods in terms of performance. In comparison to these models, SegNet further enhances performance by adopting a segmentation methodology. On one hand, it considers the closed individual interactions within the ego segment, while on the other, it also takes into account the interaction features of vehicles grouped in merging segments towards the EV. The strategy of segmenting the treatment of different vehicle interactions significantly contributes to exceptional performance.

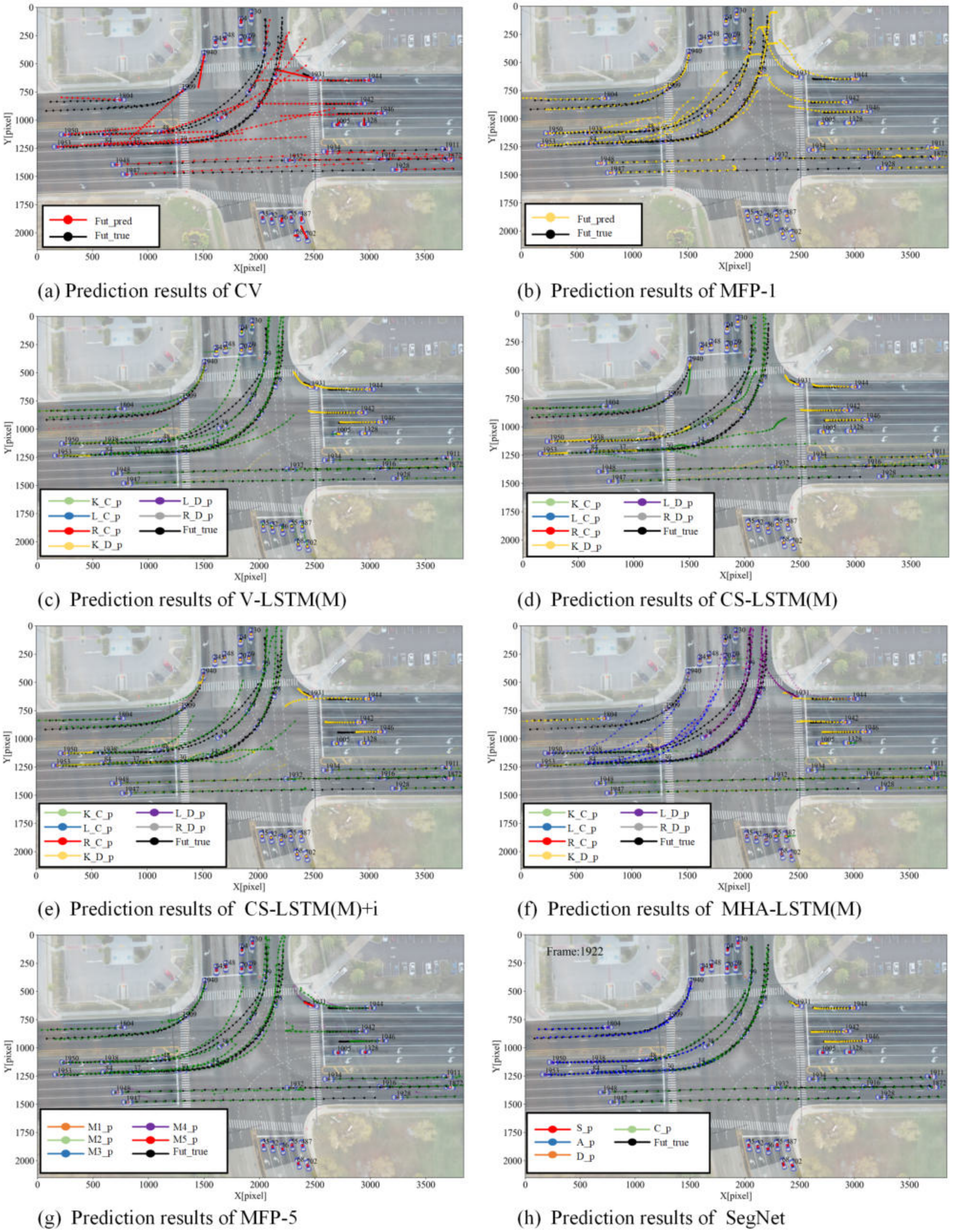
We noted that, compared to uni-modal models, multimodal prediction models exhibited significant advantages. This is most evident in the pairs of CS-LSTM and CS-LSTM (M), as well as MFP-1 and MFP-3. Once multimodal maneuvers were considered, there was a significant improvement in performance, with MFP-3 surpassing MFP-1 by 18.9%. Certainly, the significant disparity is also attributed to the inclusion of a wider variety of vehicle operations under signal control in the CitySim database, particularly the infrequent occurrence of stopping behaviors on highways. In addition, it is worth mentioning that employing more maneuver patterns does not necessarily guarantee better results. For example, CS-LSTM(M) predefines six maneuvers, while MFP-3 achieves superior performance with only three maneuvers. Furthermore, although MFP-5 exhibits some optimization compared to MFP-3, increasing the maneuver patterns K results in longer training times with minimal improvement in performance, and in some cases, even a decrease.

The distribution of longitudinal maneuvers in the dataset is shown in the lower part of Fig.9. Unlike on highways, vehicles exhibit more frequent acceleration, deceleration, and stopping behaviors in signalized intersections. Compared to segmentation models, the recognition of longitudinal maneuvers is more complex. Therefore, we independently train the SegNet only using the multimodal decoder and compare it with Hist_LSTM and Position_FC. The results show that our model effectively integrates additional features, enabling much more precise identification of acceleration, deceleration, and constant speed maneuvers.

As the prediction horizon increases, the prediction errors tend to increase accordingly. Based on the RMSE, it can be observed that MFP-3 and MFP-5 exhibit better performance in short-term prediction. However, in long-term (>3s) predictions, the performance gradually diverges between SegNets. From the NLL perspective, SegNets demonstrate a consistent advantage throughout the entire prediction horizon.

Fig.10 provides an intuitive visualization of the prediction results. The CV model performs poorly, as expected. MFP-1 shows acceptable performance in short-term predictions. However, its effectiveness diminishes as the prediction horizon increases. The multimodal predictions shown in Fig.10(c-h) present a significant improvement in the alignment with actual trajectories. Among them, V-LSTM falls short in considering vehicle interactions, resulting in significant disparities between predicted and actual results. By incorporating interaction features, both MHA-LSTM(M), CS-LSTM(M), and MFP-5 exhibit better. Unfortunately, these networks become more complex and struggle to accurately capture certain intricate vehicle features, leading to convoluted outcomes in some cases. And combining map image features, the predicted trajectories of CS-LSTM(M)+i appear smoother but still exhibit poor fitting to real trajectories. In contrast, SegNets demonstrate a clear superiority in performance.

Fig.11 also showcases multimodal prediction trajectories with probabilities in this scenario. For instance, vehicle 1942 decelerates upon detecting a red light, during which SegNets effectively utilize signal information and identify a deceleration probability of 0.99, surpassing MFP-5 and CS-LSTM(M). And the results of SegNets exhibit a close alignment with the actual trajectory. As observed in this case, the MFP model demonstrates a significant bias when confronted with variations between the learned slots. Unexpectedly, all multimodal prediction models successfully identify the deceleration maneuver for vehicle 1931. However, V-LSTM(M) and CS-LSTM(M), which neglect considerations of vehicle interactions and road geometry features in merging segment 2, still exhibit issues such as convoluted trajectories and poor alignment. And it can be noticed that CS-LSTM(M)+i demonstrates higher accuracy and better alignment with real trajectories than CS-LSTM(M). Furthermore, MHA-LSTM(M) exhibits unstable performance as it fails to provide reliable trajectory predictions for many vehicles. In contrast, both MFP-5 and SegNets demonstrate an excellent matching of trajectories, with our model exhibiting more confidence in predicting the deceleration maneuver. Similarly, vehicle 48 and 1940 provide evidence of the exceptional performance of our proposed model when incorporating road geometry features and vehicle interactions within the segment.



(a) Prediction results of CV

(b) Prediction results of MFP-1

(c) Prediction results of V-LSTM(M)

(d) Prediction results of CS-LSTM(M)

(e) Prediction results of CS-LSTM(M)+i

(f) Prediction results of MHA-LSTM(M)

(g) Prediction results of MFP-5

(h) Prediction results of SegNet

Fig. 10. Results visualization of different prediction models. In V-LSTM(M) and CS-LSTM(M), the symbols K, L, and R stand for lane-keeping, left lane change, and right lane change, respectively, while C and D indicate constant speed and deceleration. For MFP, M represents different modes. In SegNet, S, A, D, and C denote stopping, accelerating, decelerating, and maintaining constant speed, respectively.

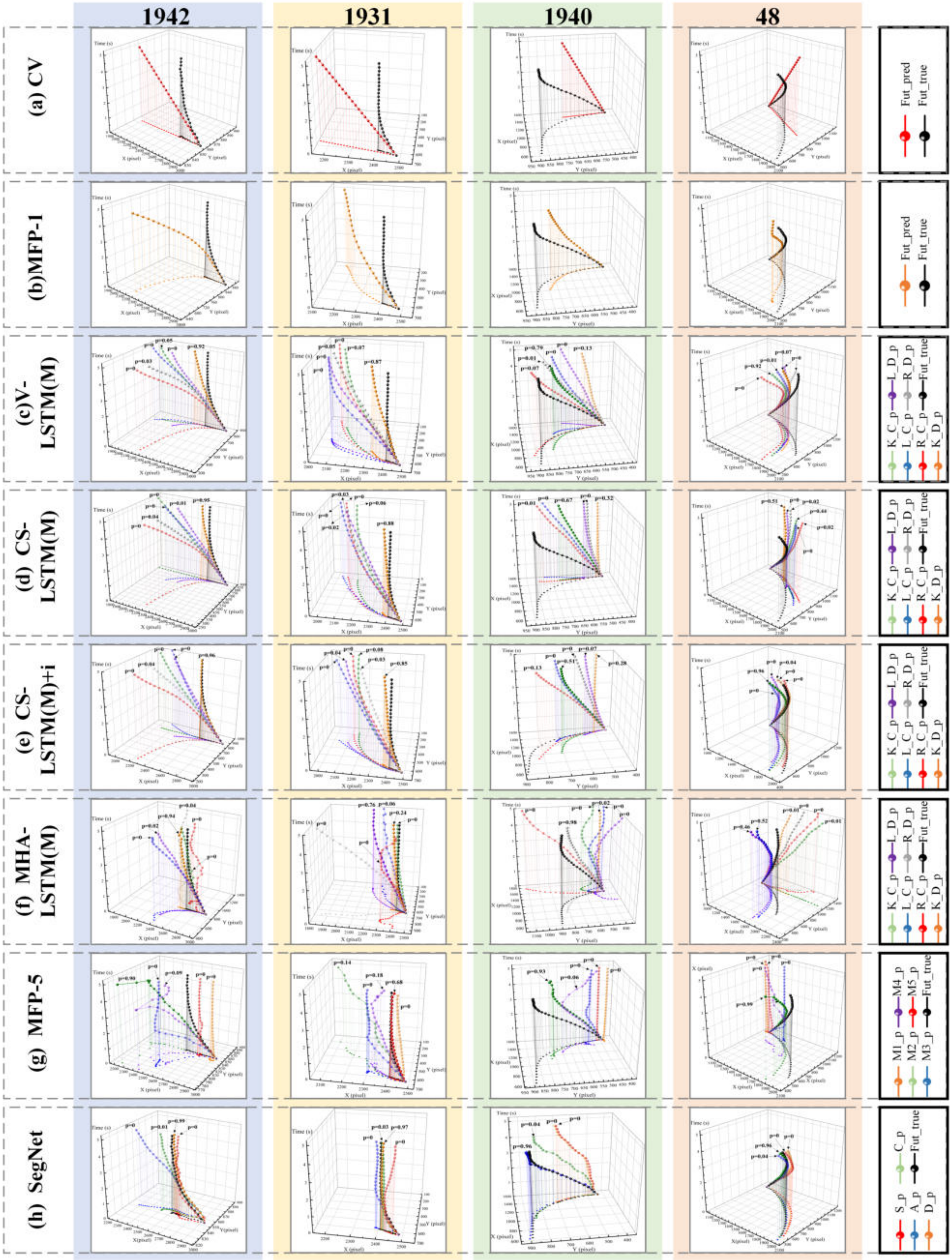


Fig. 11. multimodal prediction trajectory results of typical vehicles in scenario. Each row represents a typical vehicle, and each column represents a method. The transparency of trajectories decreases with lower probabilities (minimum 50%).

TABLE III
SIMULATION RESULTS

Metric	CV	V-LSTM(M)	CS-LSTM(M)	MFP-3	SegNets
Average jerk [m/s³]	0	3.61	7.16	9.22	4.24
Average speed [m/s]	8.78	7.56	6.59	7.80	5.80
Vehicle collision Rate	4.9%	4.7%	3.0%	3.0%	0.6%
Road collision Rate	31.2%	11.9%	7.6%	7.1%	0%
Rules violation Rate	78.1%	32.3%	23.0%	26.7%	0%
Average risk value	2.17	1.47	1.24	1.38	1.14

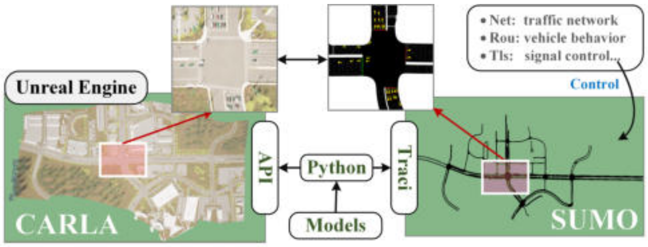


Fig. 12. CARLA-SUMO co-simulation framework

B. Real-time simulation

We carried out real-time simulation tests To validate the effectiveness of prediction models. CARLA is an outstanding open-source validation software in the field of autonomous driving [52], developed on the UE4 platform. It provides realistic environments, exceptional vehicle and sensor simulations, and flexible control interfaces, supporting a wide range of academic research [20] [53]. Importantly, to compensate for the limitations in the joint intermodal simulation of extensive road networks, CARLA has opened interfaces with SUMO, which is a powerful traffic simulation software that can simulate various aspects of urban traffic systems, including road networks, vehicle behavior, and traffic signal control [54]. Remarkably, the CitySim dataset includes CARLA maps constructed by RoadRunner and SUMO maps constructed using OpenStreetMap for an accurate representation of the road network.

Thus, we adopted the CARLA-SUMO co-simulation framework, as illustrated in Fig.12. In our simulation setup, SVs are controlled and synchronized with CARLA through the use of SUMO. The road network is generated using the net file, vehicle behavior is defined in the rou file, and traffic signals are controlled by the tls file. Vehicles use the Krauss car-following model with acceleration=2.6 m/s^2, deceleration =4.5 m/s^2, and sigma=0.5. These settings adhere to the default configurations provided by the CitySim dataset. Note that the spawned vehicles from SUMO always perceive the risk and avoid collision. This means that at each step we take, SUMO vehicles will exhibit different reactions, including collision avoidance, acceleration to pass, deceleration for yielding, and stopping, among other behaviors. So in the test, a subset of 10% of SVs were randomly generated and programmed to disregard collision avoidance within junctions. This behavior was achieved by configuring parameters such as $jmIgnoreFoeProb = 1$, $jmIgnoreFoeSpeed = 50$, and $jmIgnoreJunctionFoeProb = 1$. Meanwhile, CARLA generates the EV that are controlled by scripts. The predicted trajectory of maximum probability is employed as the input for the vehicle's control system when simulating the takeover of the CAV by RSU. In Carla, the EV planning module utilizes a quintic polynomial Parametric curve within the Longitudinal-Lateral trajectory decomposition framework. The control module employs PID control. Each model underwent 1000 iterations of testing, with data collection performed at a frame rate of 5 Hz. The next test iteration was initiated when detected a collision. The results are presented in Table.III.

Jerk is an important metric for evaluating vehicle comfort, calculated by the derivative of acceleration. The average of the three highest 1/TTC is used to assess the vehicle's risk, with lower values indicating better safety. The rules violation rate records the proportion of abnormal behaviors such as red light violations, speeding, illegal lane changes, and wrong-way driving. Focusing on vehicle kinematic features, V-LSTM (M) demonstrates commendable performance in terms of comfort. However, there are still limitations in improving efficiency and safety. CS-LSTM (M) and MFP offer their respective advantages, but their effectiveness falls short of meeting safety requirements. In comparison, SegNets exhibit the slowest average speeds as they consistently adhere to traffic signals, resulting in zero rule violations and the lowest risk coefficients. Additionally, SegNets exhibit excellent performance in terms of vehicle interactions and utilization of road geometric information.

To showcase and validate additional assistance functionalities offered by RSU, we conducted further analysis on three scenarios: left turn under signal control, merging into a right turn, and avoiding of red-light violation vehicle. In this context, the maximum value of the risk map is capped at 20, while the maximum value for the risk value is set to 1.

1) Left turn under signal control: The scenario is composed of an EV and three SVs approaching a zebra crossing, as illustrated in Fig.13. At 16s, the left-turning vehicle should decelerate upon detecting the red light. SegNets promptly identified this characteristic, while other models failed to recognize the traffic signal control information. By utilizing the predicted multimodal trajectories, the risk values for the SVs were calculated as 0.58, 0.43, and 0.46, respectively. Fig.13(a) provides an overview of the overall risk on the road.

38 seconds later, the vehicles reach the junction, as displayed in Fig.13(b). It can be intuitively observed that SegNets effectively incorporate the road's geometric structure and predict smooth trajectories, while CS-LSTM and MFP exhibit slightly inferior performance. At this moment, the leading vehicle accelerates and swiftly moves ahead of the EV,

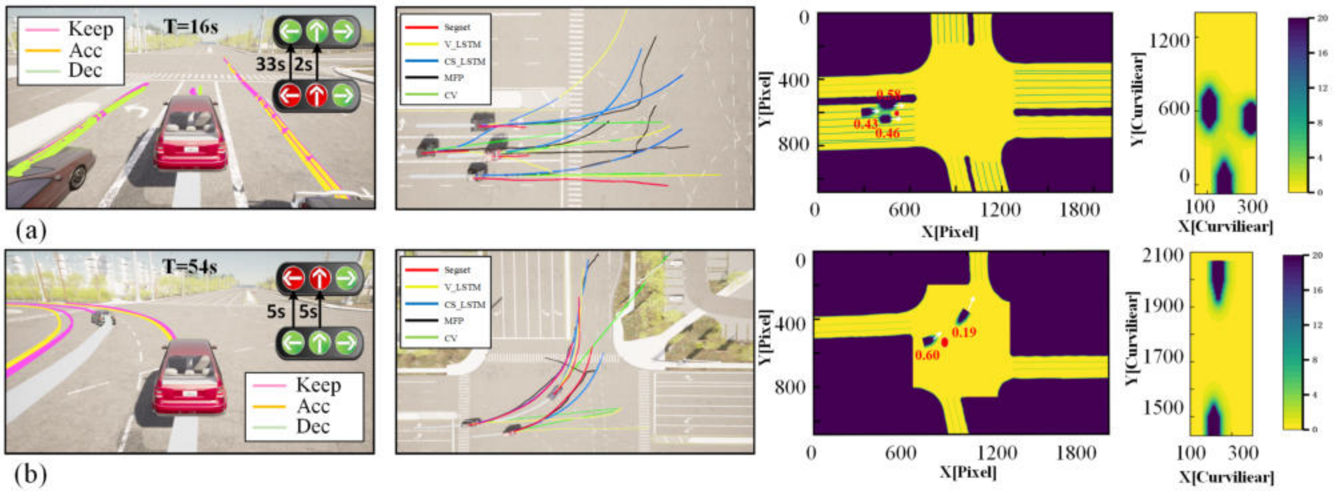


Fig. 13. Sequential display of results: SegNets multimodal outputs from vehicle view (probabilities < 0.01 excluded), comparative results from bird's eye view, risk map in pixel coordinates, and risk map in Curvilinear coordinates. (a) Test Results under red light at 16s. (b) Test Results of EV making a left turn at 54s after Green Light.

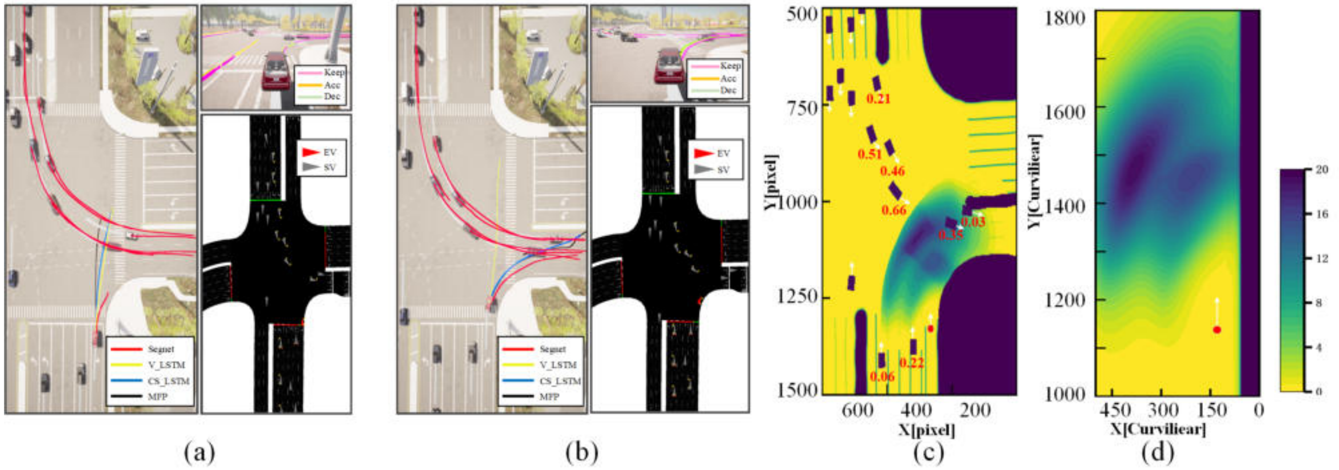


Fig. 14. Test results of Scenario 2. (a) Visualization results from different views at t=60.4. (b) Visualization results from different views at t=61.2. (c) Risk values and risk map in pixel coordinates. (d) Risk map in Curvilinear coordinates.

resulting in a low-risk value of 0.19. In contrast, the trailing vehicle gradually approaches and poses a higher threat to the EV, indicated by a risk value of 0.60. This simple scenario demonstrates the fundamental capability of our proposed architecture in considering traffic signal lights and road structure.

2) Merges into a right turn: Considering the interaction between vehicles in merging segments of a signalized intersection can be a complex scenario. Different perspectives of the scenario are illustrated in Fig.14(a)(b). At 60.4s, only SegNets successfully identify the intention of the EV to make a right turn with a probability of 0.58. It is not until 61.2 seconds that MFP and CS-LSTM finally capture the intended movement of the vehicle. Throughout the merging process, the SegNet model effectively captures the interaction information with the merging convoy. There is an 83% probability of deceleration and a 16% probability of maintaining a constant speed, highlighting its effectiveness.

Through the analysis of prediction results, we have identified the merging SV which poses the highest risk for the EV, with a risk value of 0.66. This vehicle deserves more attention, and a cautionary warning is issued accordingly. Referring to Fig.14(d), the SVs ahead on the left side of the EV are merging. Using the risk map calculation module, we evaluated the potential impact of these SVs on the EV. If the EV continues its current forward deceleration, there is a risk of encounters between SVs and the EV at the corner. Furthermore, since we assess the risk of the SVs on the EV's path, the area of convergence with the vehicle platoon remains the most critical position in terms of risk. There is a notable peak in risk level approximately 100 pixels ahead of the vehicle at the corner with a wide range of potential impact. As depicted in Fig.14(d), it is represented by two curved areas of high risk in the Curvilinear coordinate. This information can provide valuable support for trajectory planning and other related modules.

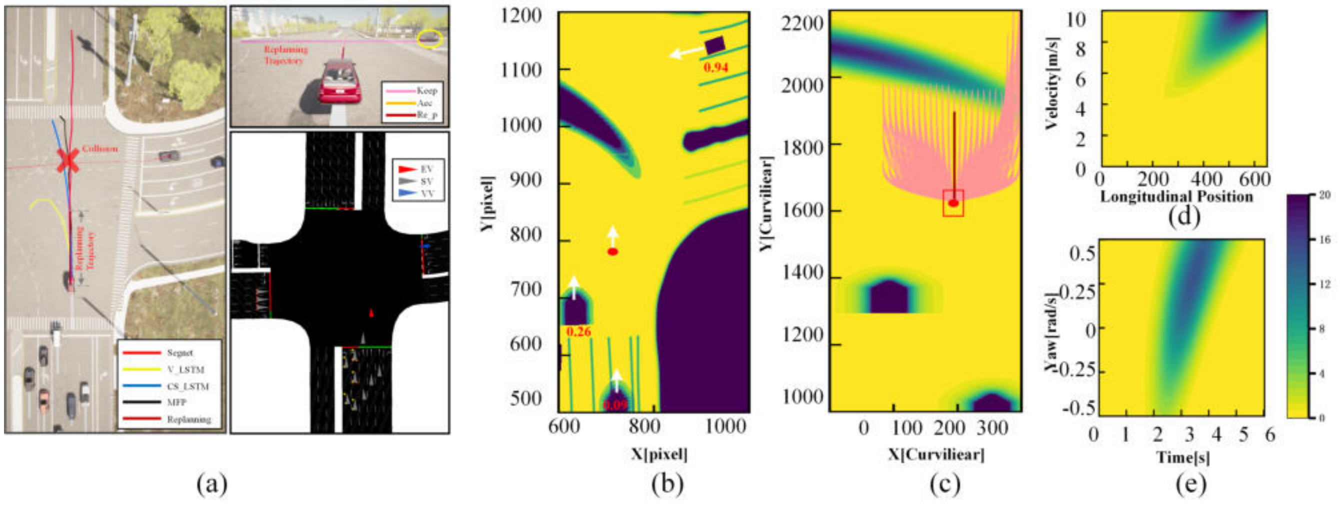


Fig. 15. Test results of Scenario 3. (a) Prediction results and re-planned trajectory from different views. (b) Risk values and risk map in pixel coordinates. (c) Risk map and replanned trajectory candidates trimming in Curvilinear coordinates. (d) Longitudinal-velocity risk map. (e) Time-Yaw risk map.

3) Avoiding a red-light violation vehicle: To validate the effectiveness of our proposed architecture in enhancing driving safety through the collision avoidance and replanning modules, we created a scenario in SUMO by setting $jmDriveAfterRedTime = 1000$ to generate a vehicle that violates a red light, as depicted in Fig.15.

Clearly, the offline CitySim database does not account for such abnormal behavior. As a result, all models, including ours, make judgments based on constant speed or acceleration, resulting in a collision.

To address this, we can incorporate a re-planning module. In Fig.15(b), the RSU identifies the violating vehicle as highly dangerous, with an attention value reaching 0.94, enabling the transmission of collision warnings to the vehicle. Additionally, the RSU can swiftly calculate more comprehensive and detailed risk zones, such as the Lateral-Longitudinal position risk map in Fig.15(d), the Longitudinal position-velocity risk map in Fig.15(e), and the Time-Yaw risk map in Fig.15(f). These varied risk maps play a crucial role in facilitating different approaches for vehicle trajectory planning and collision avoidance modules. In this scenario, these risk maps are represented by curved elliptical shapes, providing a more intuitive visualization of the hazardous areas in front of the EV if it follows its own predicted trajectory.

We employed a re-planning method described in [25]. Initially, we generated a diverse set of candidate trajectories by varying the speed, as well as lateral and longitudinal distances. Subsequently, we utilized the S-D risk map and applied a cost function to prune the candidates, thereby obtaining a feasible trajectory set. Fig.15(c)(a) illustrates the obtained optimal re-planned trajectory, represented by the deep red color. The EV followed this trajectory, successfully avoiding collision at the intersection. This result not only validates the superiority of our algorithm but also highlights the scalability of integrating different modules.

VI. Conclusion

This paper proposes a trajectory prediction and risk assessment framework to assist CAVs using the CVIS. The SegNet model is introduced to predict the future trajectories of all vehicles at a hub intersection. It divides the intersection into segments and utilizes the Curvilinear coordinate system to extract road geometric features. The model effectively utilizes individual interaction cues within the ego segment and leverages group features within the merging segment. Additionally, it incorporates valuable traffic signal information to output multimodal results. Consequently, risk value, risk map, and reference trajectory are calculated based on the multimodal prediction results. Validation results using the CitySim database and CARLA-SUMO co-simulation demonstrate that SegNet outperforms other state-of-the-art models by accurately and precisely predicting smooth trajectories that comply with traffic rules. The utilization of auxiliary information effectively helps CAVs avoid collisions and enhances driving safety.

In future research, more traffic agents will be considered to improve prediction accuracy. More comprehensive and hierarchical assistance will be introduced to enhance the driving efficiency and safety of CAVs. Ultimately, our objective is to develop a robust and versatile system capable of seamlessly adapting to diverse scenarios, encompassing a wide range of traffic conditions and road environments.

References

[1] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2023.

[2] W. M. D. Chia, S. L. Keoh, C. Goh, and C. Johnson, "Risk assessment methodologies for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16923–16939, 2022.

[3] G. Yu, H. Li, Y. Wang, P. Chen, and B. Zhou, "A review on cooperative perception and control supported infrastructure-vehicle system," *Green Energy and Intelligent Transportation*, vol. 1, no. 3, p. 100023, 2022.

[4] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[5] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 978–985, IEEE, 2021.

[6] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[7] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[8] K. Yang, X. Tang, J. Li, H. Wang, G. Zhong, J. Chen, and D. Cao, "Uncertainties in onboard algorithms for autonomous vehicles: Challenges, mitigation, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 8963–8987, 2023.

[9] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2016.

[10] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[11] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transportation research part C: emerging technologies*, vol. 99, pp. 144–163, 2019.

[12] T. Wang, S. Kim, J. Wenxuan, E. Xie, C. Ge, J. Chen, Z. Li, and P. Luo, "Deepaccident: A motion and accident prediction benchmark for v2x autonomous driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 5599–5606, 2024.

[13] D. Zhou, Z. Ma, and J. Sun, "Autonomous vehicles' turning motion planning for conflict areas at mixed-flow intersections," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 204–216, 2019.

[14] M. S. Shirazi and B. T. Morris, "Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2016.

[15] C. Chen, L. Liu, T. Qiu, Z. Ren, J. Hu, and F. Ti, "Driver's intention identification and risk evaluation at intersections in the internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1575–1587, 2018.

[16] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 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 Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1657–1666, 2021.

[18] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Advances in neural information processing systems*, vol. 32, 2019.

[19] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1468–1476, 2018.

[20] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 459–468, 2024.

[21] J. Kim and D. Kum, "Collision risk assessment algorithm via lane-based probabilistic motion prediction of surrounding vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2965–2976, 2017.

[22] D. Li, Q. Zhang, Z. Xia, K. Zhang, M. Yi, W. Jin, and D. Zhao, "Planning-inspired hierarchical trajectory prediction for autonomous driving," *arXiv preprint arXiv:2304.11295*, 2023.

[23] S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, "An improved a-star based path planning algorithm for autonomous land vehicles," *International Journal of Advanced Robotic Systems*, vol. 17, no. 5, p. 1729881420962263, 2020.

[24] W. Xihui *et al.*, "Predictive motion planning of vehicles at intersection using a new gpr and rrt," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[25] Y. Cao, W. ShangGuan, B. Cai, L. Chai, and W. Qiu, "Predictive trajectory planning for on-road autonomous vehicles based on a spatiotemporal risk field," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 1, pp. 400–420, 2022.

[26] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pp. 353–359, IEEE, 2017.

[27] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

[28] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "Pip: Planning-informed trajectory prediction for autonomous driving," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 598–614, Springer, 2020.

[29] A. Kawasaki and A. Seki, "Multimodal trajectory predictions for urban environments using geometric relationships between a vehicle and lanes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9203–9209, IEEE, 2020.

[30] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7814–7821, IEEE, 2022.

[31] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987, IEEE, 2023.

[32] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2020.

[33] M. Krüger, A. S. Novo, T. Nattermann, and T. Bertram, "Interaction-aware trajectory prediction based on a 3d spatio-temporal tensor representation using convolutional–recurrent neural networks," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1122–1127, IEEE, 2020.

[34] Z. Li, Z. Chen, Y. Li, and C. Xu, "Context-aware trajectory prediction for autonomous driving in heterogeneous environments," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 1, pp. 120–135, 2024.

[35] M. Fu, T. Zhang, W. Song, Y. Yang, and M. Wang, "Trajectory prediction-based local spatio-temporal navigation map for autonomous driving in dynamic highway environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6418–6429, 2021.

[36] T. Zhang, M. Fu, and W. Song, "Risk-aware decision-making and planning using prediction-guided strategy tree for the uncontrolled intersections," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[37] T. Puphal, M. Probst, and J. Eggert, "Probabilistic uncertainty-aware risk spot detector for naturalistic driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 406–415, 2019.

[38] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[39] F. Damerow and J. Eggert, "Predictive risk maps," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 703–710, IEEE, 2014.

[40] H. Tan, G. Lu, and M. Liu, "Risk field model of driving and its application in modeling car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11605–11620, 2021.

[41] Z. Zhou, Y. Wang, G. Zhou, K. Nam, Z. Ji, and C. Yin, "A twisted gaussian risk model considering target vehicle longitudinal-lateral motion states for host vehicle trajectory planning," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[42] R. Mahjourian, J. Kim, Y. Chai, M. Tan, B. Sapp, and D. Anguelov, "Occupancy flow fields for motion forecasting in autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5639–5646, 2022.

[43] Y. Du, Y. Shi, C. Zhao, Z. Du, and Y. Ji, "A lifelong framework for data quality monitoring of roadside sensors in cooperative vehicle-infrastructure systems," *Computers and Electrical Engineering*, vol. 100, p. 108030, 2022.

[44] A. Wu, T. Banerjee, K. Chen, A. Rangarajan, and S. Ranka, "A multi-sensor video/lidar system for analyzing intersection safety," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1158–1165, 2023.

[45] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer*

Vision – ECCV 2020 (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 194–210, Springer International Publishing, 2020.

[46] J. Mao, S. Shaoshuai, W. Xiaogang, and L. Hongsheng, "3d object detection for autonomous driving: A comprehensive survey," *International Journal of Computer Vision*, vol. 131, p. 1909–1963, Aug. 2023.

[47] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[48] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE international conference on robotics and automation*, pp. 987–993, IEEE, 2010.

[49] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *2008 IEEE International Conference on Robotics and Automation*, pp. 3731–3736, IEEE, 2008.

[50] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 341–355, 2019.

[51] O. Zheng, M. Abdel-Aty, L. Yue, A. Abdelraouf, Z. Wang, and N. Mahmoud, "Citysim: A drone-based vehicle trajectory dataset for safety-oriented research and digital twins," *Transportation Research Record*, 2023.

[52] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*, pp. 1–16, PMLR, 2017.

[53] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9610–9616, IEEE, 2019.

[54] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International journal on advances in systems and measurements*, vol. 5, no. 3&4, 2012.

Yue Cao obtained his B.S. degree from Beijing Jiaotong University, Beijing, China, in 2018, where he is currently pursuing a Ph.D. degree through the master-doctor combined program. In 2023, he served as a visiting Ph.D. student at the Informatics Institute, Faculty of Science, University of Amsterdam, Netherlands. His research focuses on trajectory prediction, decision-making, trajectory planning, and risk assessment for autonomous vehicles under the Cooperative Vehicle Infrastructure System.

Wei ShangGuan received the B.S., M.S., and Ph.D. degrees from Harbin Engineering University, in 2002, 2005, and 2008 respectively. From 2013 to 2014, he was an Academic Visitor with the University College London. He is currently a Professor and a Supervisor for PhDs studies with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include simulation and testing, integrated navigation, cooperative vehicle infrastructure system, intelligent transportation system, and train control system.

Arnoud Visser graduated in 1987 in physics at Leiden University and received a PhD in computer science at the University of Amsterdam in 2007. He has been a visiting researcher at European Space Agency and TNO. He was in the period 1995-2001 scientific advisor for the Ministry of Transport. He is currently a senior lecturer on robotics and artificial intelligence at the University of Amsterdam. His research focuses on the coöperation between intelligent systems; building a joint world model and performing shared decision making.

Junjie Chen received his Ph.D. degree in traffic information engineering and control from Beijing Jiaotong University in 2020. He was a research assistant at Carnegie Mellon University (CMU), Pittsburgh, PA, USA, from 2018 to 2020. He was a postdoctor at Tsinghua University, Beijing, China, from 2020 to 2024. He is currently an Associate Professor with the School of Automation and Intelligence, Beijing Jiaotong University. His research interests include Bayesian nonparametric learning, platoon operation control, and recognition and application of human driving characteristics.

Linguo Chai received the B.S, M.S, and Ph.D. from Beijing Jiaotong University in 2010, 2012, and 2018 respectively. Now he is an associate professor in School of Electronic and Information Engineering, Beijing Jiaotong University. From 2016 to 2017, he was a visiting scholar in UC Berkeley. His research interests include vehicle operational control of CVIS, modelling and simulation of transportation system.

BaiGen Cai received the B.S., M.S., and Ph.D. degrees from Beijing Jiaotong University, Beijing, China, in 1987, 1990, and 2010, respectively, all in traffic information engineering and control. From 1998 to 1999, he was a Visiting Scholar with Ohio State University. Since 1990, he has been on the Faculty of the School of Electronic and Information Engineering, Beijing Jiaotong University, where he is currently a Professor and the dean of the School of Computer and Information Technology. His research interests include train control system, intelligent transportation system, GNSS navigation, and intelligent traffic control.