

ORIGINAL ARTICLE

Open Access



Motion Planning for Autonomous Driving with Real Traffic Data Validation

Wenbo Chu^{1,2}, Kai Yang², Shen Li³ and Xiaolin Tang^{2*} 

Abstract

Accurate trajectory prediction of surrounding road users is the fundamental input for motion planning, which enables safe autonomous driving on public roads. In this paper, a safe motion planning approach is proposed based on the deep learning-based trajectory prediction method. To begin with, a trajectory prediction model is established based on the graph neural network (GNN) that is trained utilizing the INTERACTION dataset. Then, the validated trajectory prediction model is used to predict the future trajectories of surrounding road users, including pedestrians and vehicles. In addition, a GNN prediction model-enabled motion planner is developed based on the model predictive control technique. Furthermore, two driving scenarios are extracted from the INTERACTION dataset to validate and evaluate the effectiveness of the proposed motion planning approach, i.e., merging and roundabout scenarios. The results demonstrate that the proposed method can lower the risk and improve driving safety compared with the baseline method.

Keywords Trajectory prediction, Graph neural network, Motion planning, INTERACTION dataset

1 Introduction

Currently, autonomous vehicles are attracting more and more attention from both industry and academics due to their potential to improve traffic efficiency and safety [1]. However, the real-world traffic situation is particularly complex, and the safety issue remains a barrier to the development of autonomous vehicles [2]. In particular, safely navigating in complicated situations, such as roundabouts, is challenging for autonomous driving. If an autonomous vehicle attempts to move through a roundabout, it first ought to accurately anticipate the future trajectory of the road users nearby (e.g., vehicles and pedestrians) [3–5]. In this way, the autonomous vehicle can make safe motion planning and prevent potential

collisions with surrounding road users. Therefore, the trajectory prediction of the surrounding road users and motion planning are the two critical issues of autonomous driving [6–8]. Currently, most of the studies have studied both separately. Nevertheless, from a holistic perspective, the functions of the two are closely integrated, making it necessary to study them together.

Trajectory prediction is paramount to planning motions in the presence of other road users. However, this is a particularly difficult task because the motion of road users is multi-modal and relies on other actors. Over the past few decades, numerous algorithms have been proposed and developed in the field of trajectory prediction. Presently, motion prediction approaches can roughly be divided into three categories, namely, physics-based [9, 10], maneuver-based [11], and interaction-aware methods [12–14]. Physics-based trajectory prediction method identifies each traffic participant on the road as an independent individual and predicts its trajectory using its kinematic or dynamic features. The bulk of this type of prediction approach disregards interactions with other road users and traffic regulations, so

*Correspondence:

Xiaolin Tang
tangxl0923@cqu.edu.cn

¹ Western China Science City Innovation Center of Intelligent and Connected Vehicle, Chongqing 400044, China

² College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China

³ School of Civil Engineering, Tsinghua University, Beijing 100084, China



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

the prediction performance is not ideal [15]. Besides, the maneuver-based is to make a choice of policy and make predictions based on this policy, but this approach also ignores the interaction features between other road users [16, 17]. Recent studies display that the interaction-aware method which incorporates the interactive knowledge of driving behavior and performs better than the previous two methods has become a highly popular trajectory prediction method [18]. To forecast an agent's trajectory accurately, most interaction-aware approaches employ deep learning techniques. Recurrent Neural Networks (RNN) and Graph Neural Networks (GNN) are the two commonly used trajectory prediction neural networks. In Ref. [19], to encapsulate the dynamics of agents and represent interaction by transferring information among the agents with a predefined distance, a social long short-term memory (LSTM) model, i.e., a conventional RNN-based prediction model was presented. Nonetheless, no consideration was given to the spatial interaction of the surrounding agents. To solve this issue, another work [12], the convolutional social pooling technique was proposed. In detail, convolutional neural networks (CNN) are used to extract the interaction information between vehicles on the highway. However, for complex scenarios such as a roundabout, this method cannot accurately grasp the behavior interaction features of agents, and thus, the prediction performance is bad. In contrast to LSTM, GNN is suited for modeling interaction due to its unique nature. In Ref. [13], A GNN-RNN-based trajectory prediction method is proposed, where vehicles' dynamics features are extracted from their historical tracks using RNN, and the inter-vehicular interaction is represented by a directed graph and encoded using the graph attention networks. However, it focuses on single-agent trajectory prediction and only focuses on vehicles, without considering other types of agents such as pedestrians. To tackle this issue, an improved scheme called GRIP++ where both fixed and dynamic graphs are utilized for trajectory predictions of different types of traffic agents including vehicles and pedestrians [16]. Furthermore, a heterogeneous graph attention network is built for multi-agent trajectory prediction, which can deal with the heterogeneity of the target agents and traffic participants involved [18]. Given the effectiveness of GNN in the task of trajectory prediction, GNN will be used in this paper to provide prediction results for downstream planners. Note that the focus of this paper is not to improve forecasting accuracy, but to study how to integrate prediction results with the motion planner better.

On the other hand, the motion planning technique, as previously discussed, is another key for autonomous vehicles to accomplish autonomous driving tasks, which poses a significant impact on driving safety. Heuristic

path planning algorithms include the A* algorithm [20], the Dijkstra algorithm [21], and others [22]. This type of algorithm attempts to find a path between a set of nodes by computing the lowest cost evaluation value from each node to the destination. Because these nodes are discontinuous, they must be smoothed further before being used in practice. To achieve a smooth and dynamic feasible trajectory, optimization-based approaches such as model predictive control (MPC) [23, 24], and artificial potential field (PF) [25, 26] are commonly utilized. At present, optimization-based methods are widely applied for motion planning, especially the MPC technique. The reason is that this method could explicitly define the collision and dynamics constraints, which converts the motion planning task into the optimal control issue. While these techniques have been successfully utilized in autonomous driving demonstrations, motion planning issues remain challenging in highly interactive environments such as roundabouts and merging scenarios. Previous studies often assume other road users maintain constant velocity or follow given trajectories, which can be accurately accessed by autonomous vehicles. However, this assumption can hardly be held. In practice, the future behavior of other road users is generally required to be obtained by prediction models. It can be found that it is particularly important to combine prediction results and the motion planner appropriately. In addition, learning-based approaches, such as reinforcement learning and imitation learning [27–29], are widely used to solve the motion planning problem. However, the downside of these techniques is that it is notoriously difficult to ensure safety. Thus, this paper will adopt an optimization-based method, i.e., MPC, to design the motion planner since it is well-suited for processing the output information of the trajectory prediction model.

Currently, integrating neural network predictors with planners has gradually attracted attention. For example, a mathematical control framework based on MPC encompassing an RNN architecture was proposed [30]. Similarly, an MPC-based motion planning algorithm that incorporates a GNN predictor was built to achieve cooperation-aware lane change [31]. Recently, Ref. [32] presented an interaction-aware planner that utilizes a neural network-based predictor and analytically integrates it with MPC. However, these works mainly focus on how to improve lane change efficiency through interaction-aware prediction, and only consider vehicles, without considering pedestrians, etc. This paper tries to investigate whether neural network-based predictors could improve planning safety with real traffic data validation. The main contributions and the technical advancements are summarized as follows. A GNN network is established and trained for the trajectory prediction

of road users including vehicles and pedestrians on the INTERACTION dataset [33]. Then, hard collision avoidance constraints are designed based on the prediction results. More importantly, an MPC-based motion planning method that takes into account the collision avoidance constraints of surrounding road users is proposed. Finally, two driving scenarios, merging and roundabout scenarios are extracted and reconstructed from the INTERACTION dataset for validation and evaluation of the proposed method.

The rest of this paper is organized as follows. Section 2 presents the overall architecture of the motion planning framework with the GNN-based trajectory prediction model. Section 3 demonstrates the specific process of GNN-based motion prediction. Section 4 provides the design for the MPC-based motion planning approach. The INTERACTION dataset is used to evaluate the proposed motion prediction model, and two cases are carried out to verify the proposed motion planning method in Section 5. The conclusion and future work are given in Section 6.

2 Motion Planning Framework with Graph Neural Network-Based Prediction Model

Figure 1 demonstrates the conceptual framework of motion planning with the GNN-based trajectory prediction model in this paper. The framework is mainly composed of two layers, including the trajectory prediction part, and the motion planning part that plans motion based on the predicted future trajectories of the surrounding road users. Firstly, to accurately capture the interaction features among road users, the GNN is utilized in this work. In detail, each road user is modeled as a node in the graph. The historical dynamic information

of road users, i.e., position, velocity, and heading angle are processed by the RNN encoder, which is also used as the node features. The interaction features are represented as edge attributes among nodes. Then, the dynamic features and interaction features are fed to the LSTM decoder to generate the final prediction trajectory. To achieve safe motion planning, the MPC technology is utilized to handle the predicted trajectory. Specifically, collision avoidance constraints are designed based on the future trajectory of surrounding road users. In addition, the traffic rules including speed limits, and road boundary limits are also considered. More importantly, the target driving path, target speed, and control efforts are processed in the objective function. In this way, the motion planning issue could be converted into an optimization problem with multiple constraints. Finally, two challenging cases, including roundabout and merging scenarios are extracted from the realistic driving logs, i.e., the INTERACTION dataset.

3 Graph Neural Network-Based Trajectory Prediction Model

The driving environment is dynamic, and it is crucial for autonomous vehicles to accurately predict the future motion of nearby road users. This section aims to establish a prediction model that could simultaneously forecast the future trajectory of surrounding road users near the ego vehicle (the ego vehicle is referred to as the controlled autonomous vehicle).

According to Refs. [16, 18], the GNN is suitable to model and process the interaction among road users. Each road user is modeled as a node in the GNN. A directed graph is usually represented by $\mathcal{G} = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of n nodes (assume

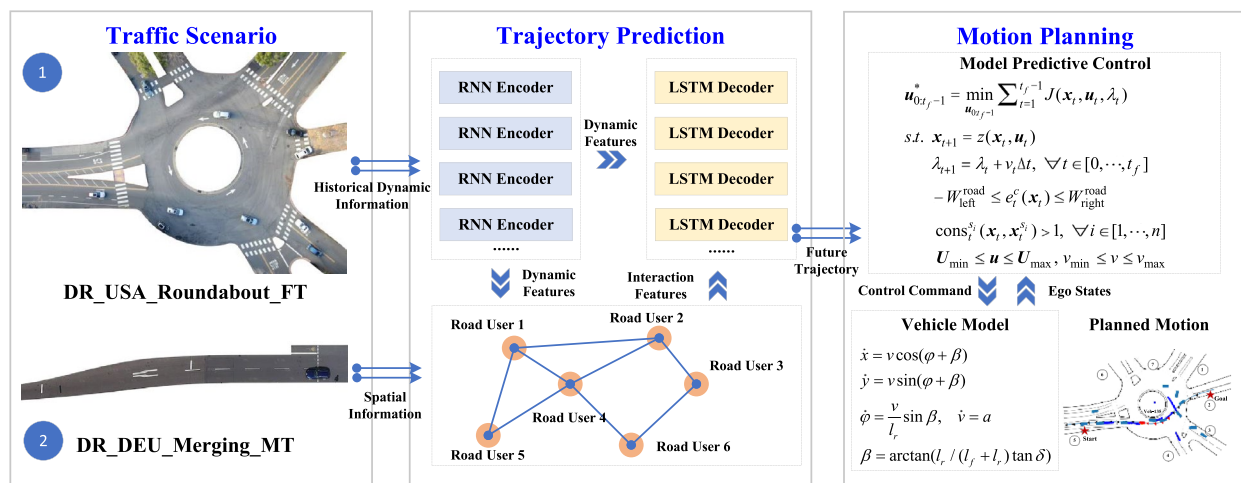


Figure 1 The motion planning framework with graph neural network-based prediction model

that there are n road users that need to be predicted), and $E \subset V \times V$ is the set of directed edges. Each node contains its node feature, and each directed edge contains an edge feature.

The input to the prediction model is X_t , which is composed of the trajectory history of all predicted road users in the target region.

$$X_t = [\mathbf{h}_t^1, \mathbf{h}_t^2, \dots, \mathbf{h}_t^n], \tag{1}$$

where X_t is a collection of historical state information for all road users; $\mathbf{h}_t^i = [\mathbf{S}_{t-t_h}^i, \mathbf{S}_{t-t_h+1}^i, \dots, \mathbf{S}_t^i]$ represents the historical states of the road user i at time t ; $\mathbf{S}_t^i = [x_t^i, y_t^i, \varphi_t^i, v_t^i]$ contains positions x , y , heading angle φ , and velocity v of the road user i , and t_h represents the observation horizon.

The output of the prediction model is the future position of all agents, which is denoted as

$$Y_t = [f^1, f^2, \dots, f^n], \tag{2}$$

where Y_t is the set of future state information of all agents; $f^i = [(x_{t+1}^i, y_{t+1}^i), (x_{t+2}^i, y_{t+2}^i), \dots, (x_{t+t_f}^i, y_{t+t_f}^i)]$ is the future trajectory of road user i ; t_f means prediction horizon.

The details of the implementation are listed below. To begin with, the historical states of each road user need to be processed to extract the dynamic features. Currently, the RNN is commonly utilized to process the sequence information, which is utilized to encode the historical information of each road user.

$$\tilde{\mathbf{h}}^i = \text{RNN}_{\text{hist}}(\mathbf{h}_t^i), \tag{3}$$

where r_t^i is the extracted dynamic feature of the road user i at the time t . The dynamic feature r_t^i also serves as the node i 's features, which are denoted as

$$\mathbf{H}_t = [\tilde{\mathbf{h}}^1, \tilde{\mathbf{h}}^2, \dots, \tilde{\mathbf{h}}^n]. \tag{4}$$

Therefore, the node feature has been obtained via Eq. (4), i.e., each road user's dynamic feature.

The edge features are defined as follows. An edge from node j to node i means that the node i 's behavior is influenced by the node j and the edge feature is the relative measurement of the node j to node i , e.g., position, velocity v , and yaw angle φ . A set of edge attributes is denoted by:

$$E_t = \{e_{ij} | i, j \in [1, n]\}, \tag{5}$$

$$e_{ij} = [x_t^j, y_t^j, \varphi_t^j, v_t^j] - [x_t^i, y_t^i, \varphi_t^i, v_t^i]. \tag{6}$$

The output of the GNN layer is a new set of node features:

$$\bar{\mathbf{H}} = [\bar{\mathbf{h}}^1, \bar{\mathbf{h}}^2, \dots, \bar{\mathbf{h}}^n] = \text{GNN}(\mathbf{H}_t, E_t). \tag{7}$$

Currently, there are two variants of GNN, namely, the graph convolutional network (GCN) model and the graph attention network (GAT) model [17]. The GAT model, which is based on attention mechanisms, can consider the degree of influence of different road users on the predicted one. According to Ref. [18], the attention score of the node j to node i can be calculated as:

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(\bar{a}^T [\tilde{\mathbf{h}}_t^i || e_{ij} || \tilde{\mathbf{h}}_t^j]))}{\sum_{k \in N_i} \exp(\text{LeakyRelu}(\bar{a}^T [\tilde{\mathbf{h}}_t^i || e_{ij} || \tilde{\mathbf{h}}_t^k]))}, \tag{8}$$

where \bar{a} is a single-layer feed-forward neural network; LeakyRelu is the activation function; N_i means the neighbors of the node i . Thus, the feature of the node i is updated by calculating a weighted sum of edge-integrated node features over its surroundings.

$$\bar{\mathbf{h}}^i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} \text{FC}([e_{ij} || \tilde{\mathbf{h}}_t^j]) \right), \tag{9}$$

where σ is the sigmoid function, and the symbol $||$ means the concatenation operator, FC is the fully connected layer.

Finally, the LSTM-based decoder is utilized to predict the final future trajectory of the road user i by considering individual dynamics $\bar{\mathbf{h}}^i$ and its interaction with other road users $\bar{\mathbf{h}}$, i.e.,

$$f^i = \text{LSTM}_{\text{fut}}([\bar{\mathbf{h}}^i || \bar{\mathbf{h}}]). \tag{10}$$

4 GNN Prediction Model Enabled Motion Planner Based on MPC

4.1 Vehicle Kinematics Model

To depict the motion of the controlled autonomous vehicle, i.e., ego vehicle, a kinematic bicycle model (front-wheel-only) is employed while considering the computing cost and modeling accuracy. The kinematic equations are formulated as follows [34]:

$$\begin{cases} \dot{x} = v \cos(\varphi + \beta), \\ \dot{y} = v \sin(\varphi + \beta), \\ \dot{\varphi} = \frac{v}{l_r} \sin \beta, \\ \dot{v} = a, \\ \beta = \arctan \left(\frac{l_r}{l_f + l_r} \tan \delta \right), \end{cases} \tag{11}$$

where β is the slip angle; l_f and l_r denote the distance from the front and rear axles of the vehicle to the center of gravity, respectively; x and y mean the positions; v

indicates the speed along the vehicle body direction; φ is the heading angle. Set the state variable $\mathbf{x} = [x, y, \varphi, v]^T$ and control input $\mathbf{u} = [a, \delta]^T$, a is the acceleration, and δ is the steering angle. Thus, the vehicle model depicted in Eq. (11) can be briefly denoted $\dot{\mathbf{x}} = z(\mathbf{x}, \mathbf{u})$ for the subsequent discussion.

4.2 Collision Avoidance Constraints Design Considering Different Road User's Type

When operating on public roads, the autonomous vehicle must avoid collisions with other road users. In this paper, two main types of road users are considered, namely, vehicles and pedestrians. For different types of road users, we need to design different constraints. Therefore, we need to self-adapt the constraints according to their types. For example, pedestrians are the type of road users with low walking speeds, but high behavioral uncertainty.

First of all, a set of circles are utilized to approximate the occupied region of the ego vehicle, i.e., $O_{ego}(\mathbf{x}_t) = \bigcup_{i \in \{1, 2, \dots, n\}} O_i$, $\mathbf{x}_t = [x, y, \varphi, v]^T$ is the ego vehicle's states at the time t , and O_i means the area of the circle with a given radius r_{ego} . The number of required circles and their radius should be selected according to the shape parameters of the ego vehicle, and refer to Refs. [35, 36] for details (in this paper, $n = 3$). Similarly, for the other road users, we use the equation of an ellipse to describe the area they occupy. The key parameters, including the semi-major axis and the semi-minor axis, will be adjusted based on the road users' shape parameters and their types. In addition, the orientation of road users should also be considered. The rotation matrix (the counter-clockwise rotation angle is positive) is represented as:

$$\mathbf{R}(\varphi) = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix}. \quad (12)$$

Specifically, if a road user i 's state at the time t is $\mathbf{x}_t^{s_i} = [x^{s_i}, y^{s_i}, \varphi^{s_i}]^T$, the collision avoidance constraints can be denoted as:

$$\begin{cases} \text{cons}_t^{s_i}(\mathbf{x}_t, \mathbf{x}_t^{s_i}) = \\ \left[\begin{array}{c} x - x^{s_i} \\ y - y^{s_i} \end{array} \right]^T \mathbf{R}(\varphi^{s_i})^T \begin{bmatrix} \frac{1}{\mu^2} & 0 \\ 0 & \frac{1}{\eta^2} \end{bmatrix} \mathbf{R}(\varphi^{s_i}) \begin{bmatrix} x - x^{s_i} \\ y - y^{s_i} \end{bmatrix} > 1, \\ \mu = a_{s_i} + vT_0 + r_{ego} + \varepsilon, \eta = b_{s_i} + r_{ego} + \varepsilon, \end{cases} \quad (13)$$

where \mathbf{x}_t means the ego vehicle's states at the time t , including x and y ; $\mathbf{x}_t^{s_i} = [x^{s_i}, y^{s_i}, \varphi^{s_i}]^T$ is the road user i 's state at the time t , where x^{s_i}, y^{s_i} are the positions; φ^{s_i} is the heading angle; \mathbf{R} is the rotation matrix; a_{s_i} and b_{s_i} are the length and width of the vehicle i ; T_0 is the time headway factor; ε is a smaller positive value; $\forall t \in [0, \dots, t_f]$ and $\forall i \in [1, \dots, n]$. Other road users' positions on the horizon

$[0, t_f]$ are given by the prediction model via Eq. (10), and the heading angle φ^{s_i} can be obtained by differentiation operations. Furthermore, considering that pedestrians have high behavioral uncertainty, it would be better to use the circle to describe the area they occupy. Therefore, if the road user's type is pedestrian, set $a_{s_i} = b_{s_i}$.

4.3 Traffic Rules Constraints and Actuator Saturation Constraints

Traffic Rules Constraints: In general, autonomous vehicles need to obey traffic rules on public roads, such as speed limits. The speed constraint is set as

$$v_{\min} \leq v \leq v_{\max}, \quad (14)$$

where v_{\min} and v_{\max} represent the minimal and maximum allowed vehicle speeds, respectively.

In addition, autonomous vehicles must not deviate from road boundaries, i.e., off-road. To make sure the vehicle stays inside the lines of the road, the limit on the lateral offset distance from the reference path (i.e., contour error) of the autonomous vehicle is imposed, i.e.,

$$-W_{\text{left}}^{\text{road}} \leq e_t^c(\mathbf{x}_t) \leq W_{\text{right}}^{\text{road}}, \quad (15)$$

where $W_{\text{left}}^{\text{road}}$ and $W_{\text{right}}^{\text{road}}$ indicate the boundaries of the road.

Actuator Saturation Constraints: The input variables should be restricted to consider the actuator saturation.

$$\mathbf{U}_{\min} = \begin{bmatrix} a_{\min} \\ \delta_{\min} \end{bmatrix} \leq \mathbf{u} = \begin{bmatrix} a_t \\ \delta_t \end{bmatrix} \leq \begin{bmatrix} a_{\max} \\ \delta_{\max} \end{bmatrix} = \mathbf{U}_{\max}, \quad (16)$$

where a_{\min} and a_{\max} represent the minimal and maximum allowed acceleration that the vehicle can reach. δ_{\min} and δ_{\max} are the minimal and maximum steering angles, respectively.

4.4 MPC-Based Motion Planner Design

Objective function: we assume that there is a high-level planner that gives a reference velocity v_{ref} and reference path. Because, a route planner, usually, exists for autonomous driving systems to give a reasonable global driving path and optimal driving speed based on the current traffic flow. Then, the objective function is defined as follows to allow the autonomous vehicle to track the reference velocity while following the reference path:

$$\begin{aligned} J(\mathbf{x}_t, \mathbf{u}_t, \lambda_t) = & \|e_t^c(\mathbf{x}_t, \lambda_t)\|_{q_c} + \\ & \|e_t^l(\mathbf{x}_t, \lambda_t)\|_{q_l} + \|v_{ref} - v_t\|_{q_v} + \|a_t\|_{q_a} + \|\delta_t\|_{q_\delta}, \end{aligned} \quad (17)$$

where e_t^c and e_t^l represent contour error and lag error, respectively. The contour error gauges the ego vehicle's

lateral departure from the reference path, whereas the lag error quantifies the longitudinal displacement of the ego vehicle from the reference path, and for details, refer to Ref. [35]. In addition, λ_t represents the ego vehicle's current progress on the reference path. $q_c, q_l, q_v, q_a, q_\delta$ are the weight factors. At each stage, the MPC planner calculates the optimal control command to minimize the current cost.

With a prediction horizon t_f , the motion planning problem is constructed as a receding horizon trajectory optimization problem. The optimization issue is finally established as follows:

$$\begin{aligned}
 \mathbf{u}_{0:t_f-1}^* &= \min_{\mathbf{u}_{0:t_f-1}} \sum_{t=1}^{t_f-1} J(\mathbf{x}_t, \mathbf{u}_t, \lambda_t), \\
 \text{s.t., } \mathbf{x}_{t+1} &= z(\mathbf{x}_t, \mathbf{u}_t), \\
 \lambda_{t+1} &= \lambda_t + v_t \Delta t, \forall t \in [0, \dots, t_f], \\
 -W_{\text{left}}^{\text{road}} &\leq e_t^c(\mathbf{x}_t) \leq W_{\text{right}}^{\text{road}}, \\
 \text{cons}_t^{s_i}(\mathbf{x}_t, \mathbf{x}_t^{s_i}) &> 1, \forall i \in [1, \dots, n], \\
 v_{\min} &\leq v \leq v_{\max}, \\
 \mathbf{U}_{\min} &\leq \mathbf{u} \leq \mathbf{U}_{\max},
 \end{aligned} \tag{18}$$

where Δt is the control frequency; \mathbf{U}_{\min} and \mathbf{U}_{\max} are the lower and upper bounds of the control variables, corresponding to Eq. (16). So far, the motion planning problem has been transformed into the optimal control problem in the predictive time domain. The optimal problem is solved by the FORCESPRO solver [37]

5 Experiment Results and Discussion

Three cases are constructed based on the INTERACTION dataset for validation and evaluation. The ADE@3s of the prediction algorithm for the roundabout and merging scenario is 0.278 m and 0.320 m, respectively. In addition, collision avoidance with surrounding vehicles in the merging scenario, collision avoidance with surrounding pedestrians in the roundabout scenario, and collision avoidance with surrounding vehicles in the roundabout scenario are constructed.

5.1 Experiment Settings

During the experiment, to reproduce the original traffic scene in the dataset as much as possible, one of the vehicles in the selected scene is replaced to test the effectiveness of the proposed motion planner. The original driving path of the replaced vehicle in this scene segment is used as the reference path of the autonomous vehicle, and the maximum driving speed of the replaced vehicle is used as the target driving speed. Note that this is fundamentally different from the trajectory tracking control issue. The proposed planner does not need to track the speed trajectory of the vehicle being replaced. Instead, the planner

needs to achieve obstacle avoidance by controlling acceleration and steering angle based on the prediction results of the prediction model. The autonomous vehicle is allowed to deviate from the reference trajectory within a certain range.

To evaluate the performance of the proposed method, the following metric is utilized, i.e., driving safety. In detail, a driving safety metric that determines whether the ego vehicle is safe is obtained by detecting whether the optimal solution to the optimization problem (18) can be solved correctly at each time. This can be achieved via the solver, i.e., the exit_flag of the FORCESPRO solver at each time step. The reason is that collision avoidance is processed as a hard constraint in the optimization (18). In other words, if the optimal solution can be found, then it means that this constraint can be satisfied. Table 1 summarizes the parameters of the autonomous vehicle and the controller. In addition, the proposed method, called GNN-MPC is also compared to the MPC with the constant-velocity prediction model (Referred to as CV-MPC).

5.2 Case 1: Collision Avoidance with Surrounding Vehicles in the Merging Scenario

Figures 2 and 3 show a merging scenario in which the ego vehicle needs to safely cross a road that changes from two-lane to one-lane. This case is extracted from the INTERACTION dataset, i.e., the DR_USA_Roundabout_FT-vehicle_track_14 file. In this case, vehicle #19 in the DR_USA_Roundabout_FT-vehicle_track_14 file is replaced by the motion planners utilized in the paper. Specifically, its driving path is set as the reference path for the planners. And the planners need to predict the future trajectories of surrounding road users to decide the acceleration and steering angle. The desired driving velocity is set as 8 m/s (i.e., $v_{\text{ref}} = 8$ m/s) in this case, and the distance that the ego vehicle can deviate from the reference path is set to 0.5 m (i.e., $W_{\text{left}}^{\text{road}} = W_{\text{right}}^{\text{road}} = 0.5$ m).

Table 1 Parameters of autonomous vehicles and the planner

Parameter	Value	Parameter	Value
t_h (s)	1	δ_{\max} (rad)	0.22π
t_f (s)	3	v_{\min} (m/s)	0.00
l_f (m)	1.33	v_{\max} (m/s)	30.00
l_r (m)	1.81	q_c	130
Δt (s)	0.10	q_l	130
a_{\min} (m/s ²)	-8.00	q_v	4
a_{\max} (m/s ²)	2.80	q_a	6
δ_{\min} (rad)	-0.22π	q_δ	100

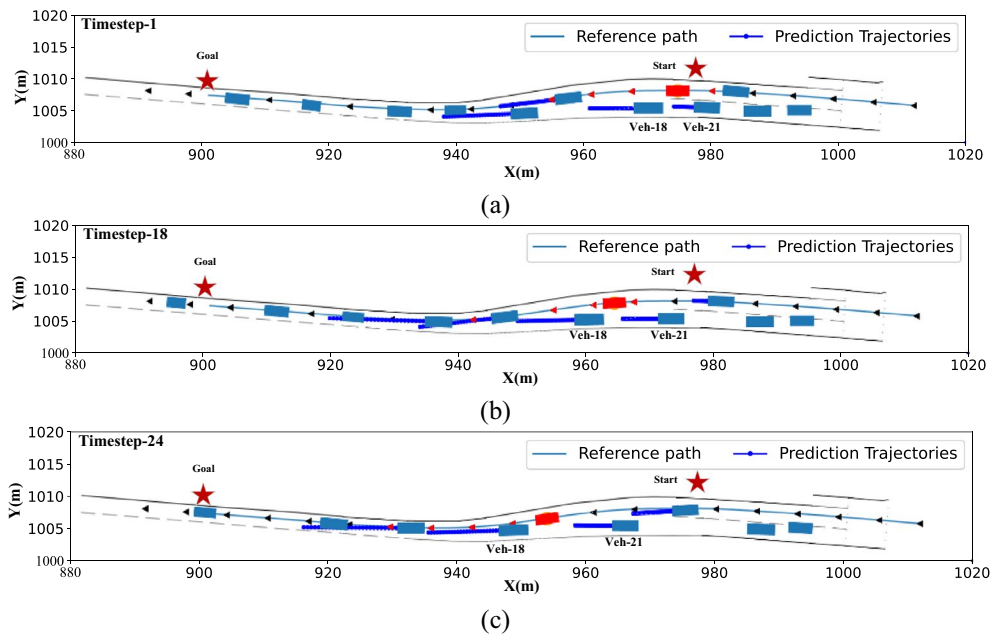


Figure 2 The motion planning results of CV-MPC for vehicle avoidance in Case 1, and (a–c) demonstrate three keyframes, i.e., timestep-1, timestep-18, timestep-24

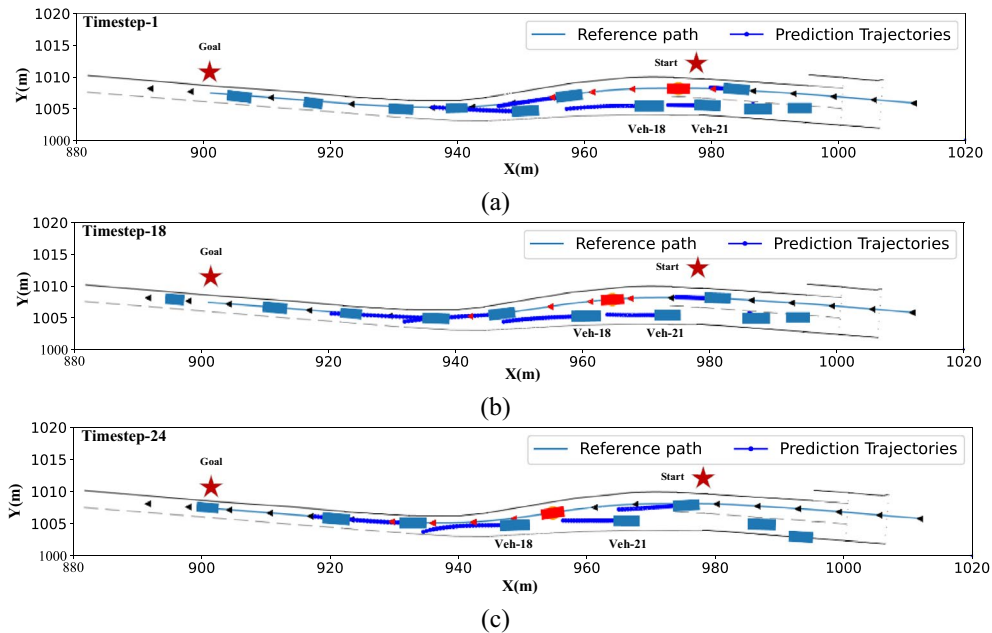


Figure 3 The motion planning results of GNN-MPC for vehicle avoidance in Case 1, and (a–c) demonstrate three keyframes, i.e., timestep-1, timestep-18, timestep-24

Three keyframes of the planned results outputted by CV-MPC and GNN-MPC planners are given in Figures 2 and 3, respectively. From the results, it can be found that road user vehicle #18 and vehicle #21 are the most significant agents affecting the safe completion of the

merging behavior of the ego vehicle. The CV-MPC and GNN-MPC both predict that vehicle #21 will yield the ego vehicle (the red vehicle in the figures). Therefore, they choose to insert the ego vehicle between vehicle #18 and vehicle #21 to complete the merging maneuver.

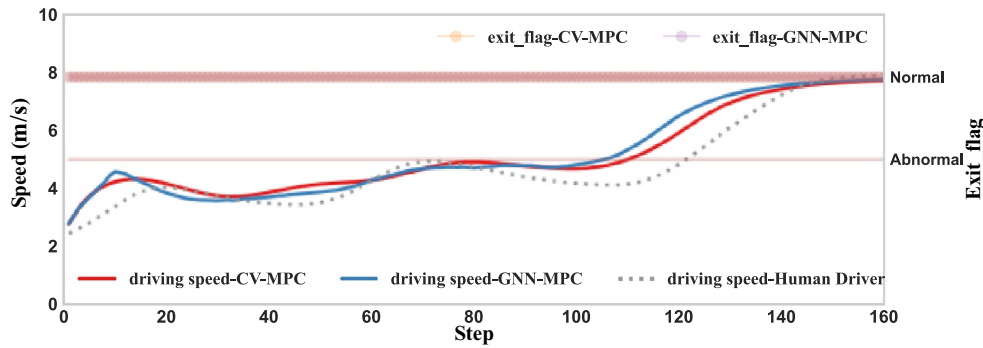


Figure 4 Comparison of driving velocity and safety performance in Case 1

The driving velocities of CV-MPC, GNN-MPC, and the human driver (i.e., the original velocity profile of the vehicle #19 in the dataset) are given in Figure 4. More importantly, the safety metric, namely, the exit_flag of the FORCESPRO solver is also shown. It can be discovered that the exit_flag at each time step is normal for both planners. In other words, collision avoidance constraints can be satisfied. Furthermore, acceleration and steering angle are demonstrated in Figure 5, which means actuator saturation constraints are also met. In summary, these results show that both CV-MPC and GNN-MPC can handle this case successfully. Note that this case is primarily intended to verify the validity of the CV-MPC and GNN-MPC planners. In this case, the two key traffic participants vehicle #18 and vehicle #21 perform straight-line driving, and both planners can make good predictions about their behavior.

5.3 Case 2: Collision Avoidance with Vehicles and Pedestrians in the Roundabout Scenario

Roundabout is a complex scene for autonomous vehicles to make safe planning. When driving in the roundabout scenario, the future trajectory of surrounding road users needs to be accurately predicted. Figures 6, 7, 8 and 9

demonstrate a more challenging case in which the ego vehicle needs to go through a roundabout while avoiding collision with other vehicles and pedestrians from the start point to the goal. This case is extracted from the INTERACTION-DR_USA_Roundabout_FT-vehicle_tracks_30 file. In this case, vehicle #125 in the INTERACTION-DR_USA_Roundabout_FT-vehicle_tracks_30 file is replaced by the motion planners utilized in the paper. Similarly, its driving path is set as the reference path for the planners. And the planners need to predict the future trajectories of surrounding road users to decide the acceleration and steering angle. The desired driving velocity is set as 5 m/s (i.e., $v_{ref} = 5$ m/s) in this case, and the distance that the ego vehicle can deviate from the reference path is set to 0.5 m (i.e., $W_{left}^{road} = W_{right}^{road} = 0.5$ m).

In this case, when the ego vehicle (the red one) enters the roundabout, it first needs to avoid vehicle #114, and it needs to accurately predict the future trajectory of vehicle #102. Figures 6a and 7a show the prediction results of CV-MPC and GNN-MPC for vehicle #102 at timestep 10, respectively. It can be found that CV-MPC predicts vehicle #102 heading to exit #7, while GNN predicts vehicle #102 heading to exit #1. Actually, it can be discovered that vehicle #102 is heading to exit

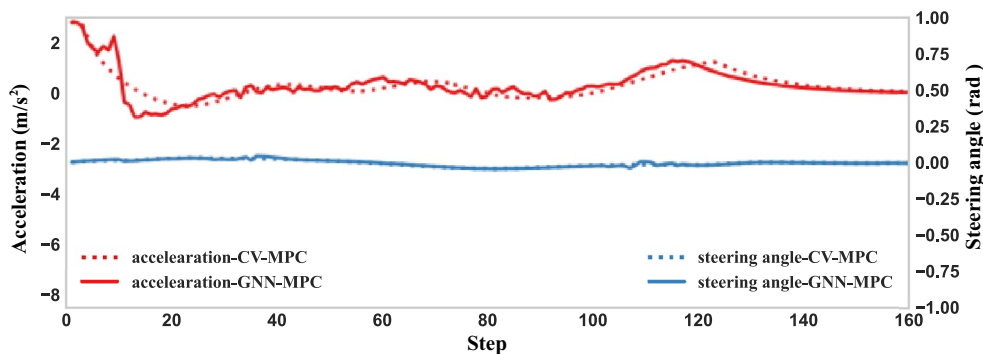


Figure 5 Comparison of acceleration and steering angle in Case 1

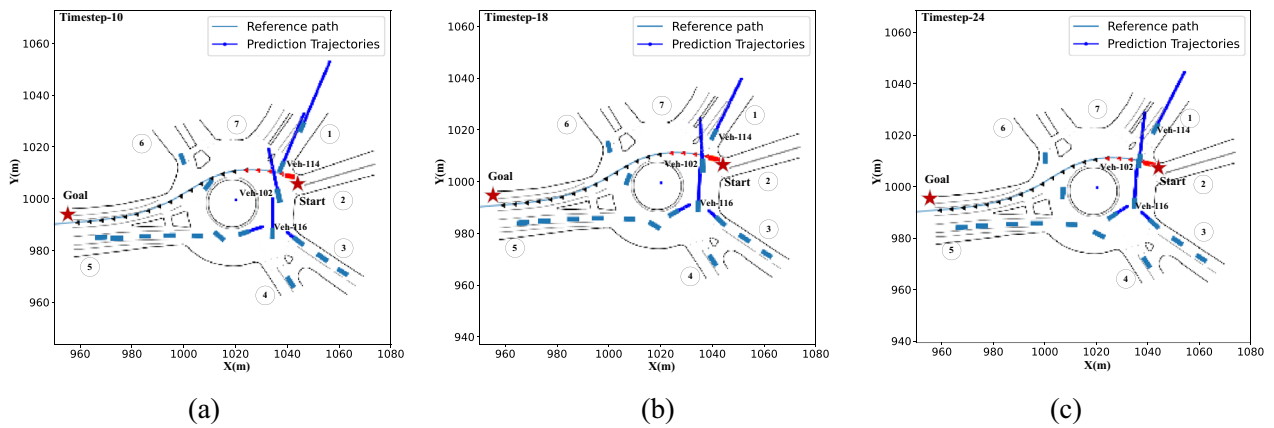


Figure 6 The motion planning results of CV-MPC for vehicle avoidance in Case 2, and (a–c) demonstrate three keyframes, i.e., timestep-10, timestep-18, timestep-24

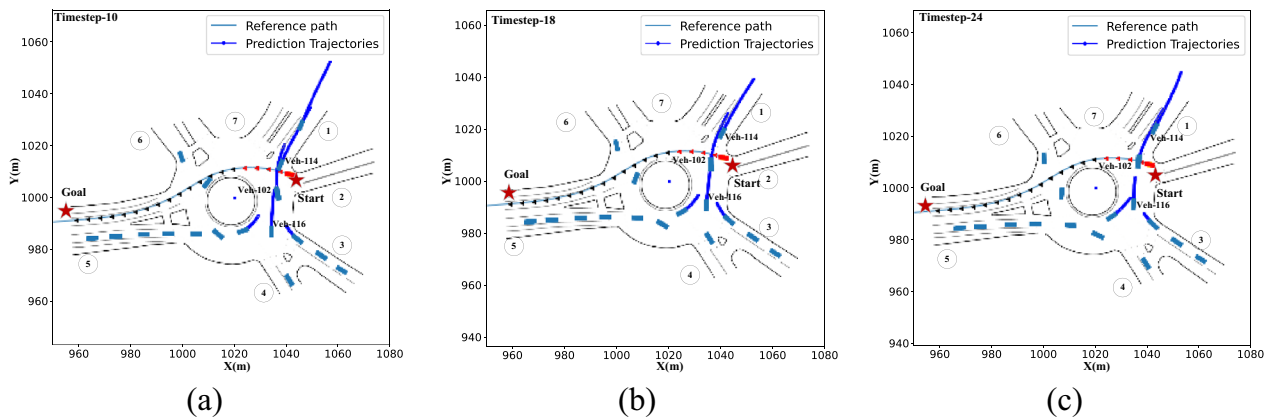


Figure 7 The motion planning results of GNN-MPC for vehicle avoidance in Case 2, and (a–c) demonstrate three keyframes, i.e., timestep-10, timestep-18, timestep-24

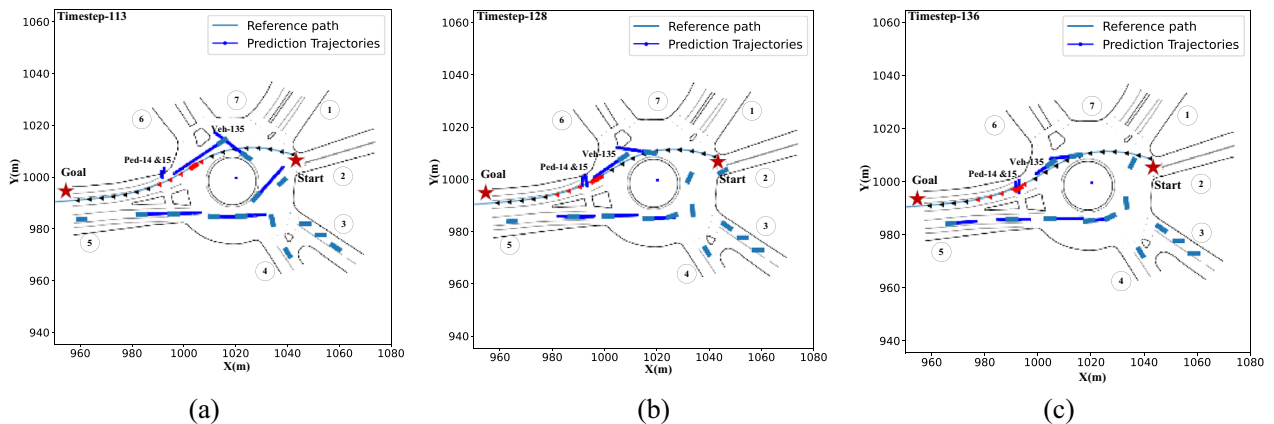


Figure 8 The motion planning results of CV-MPC for pedestrian avoidance in Case 2, and (a–c) demonstrate three keyframes, i.e., timestep-113, timestep-128, timestep-136

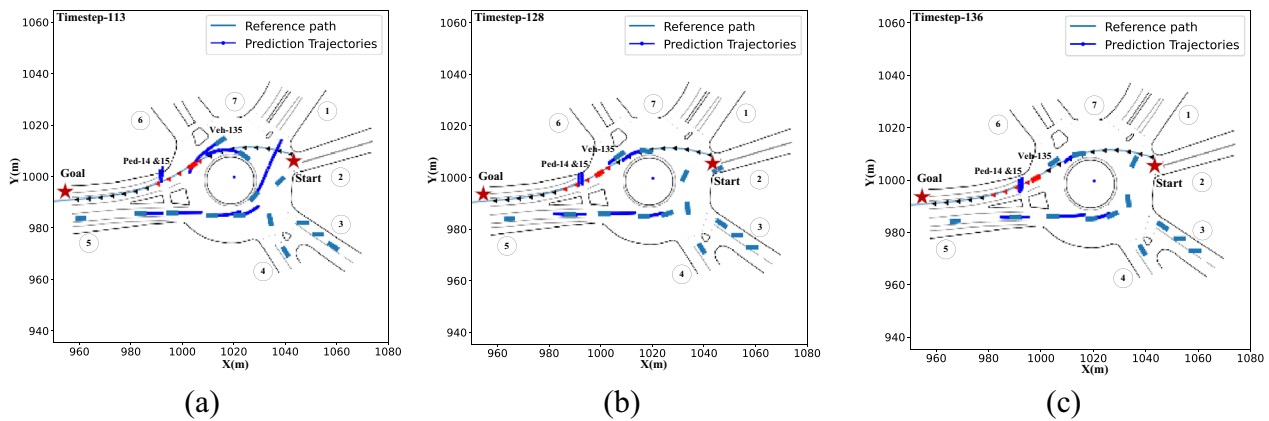


Figure 9 The motion planning results of GNN-MPC for pedestrian avoidance in Case 2, and (a–c) demonstrate three keyframes, i.e., timestep-113, timestep-128, timestep-136

#1. From Figure 6c, due to the prediction error, the ego vehicle controlled by the CV-MPC planner almost collides with vehicle #102 at timestep 24. This is consistent with the results provided in Figure 10, where the solver cannot exit properly at timestep 20–32, i.e., the collision constraint cannot be satisfied. Instead, because GNN-MPC can accurately predict the future trajectory of vehicle #102, it can plan a safe motion to avoid collisions.

Figures 8 and 9 show the planning result of avoiding pedestrians. In this scene, there are two pedestrians #14 and #15 in front of the ego vehicle. From Figure 8a, it can be found that CV-MPC predicts that the pedestrians will yield the ego vehicle at timestep 113, i.e., the ego vehicle can drive away from this area before the pedestrians reach the conflict point. However, by the time the ego vehicle reaches the conflict area, the pedestrians are approaching the conflict point, as shown in Figure 8b. Figures 8c and 10 demonstrate that the CV-MPC chooses to accelerate away, which puts

the vehicle at risk of collision with pedestrians. For the GNN-MPC, as shown in Figures 9 and 10, it predicts the pedestrians will not yield the ego vehicle and selects to decelerate to avoid collision with them. Finally, the control inputs are given in Figure 11, which shows that the acceleration and steering angle can be limited to the desired range.

5.4 Case 3: Collision Avoidance with Vehicles in the Roundabout Scenario

Figures 12 and 13 show another collision avoidance case with multiple surrounding vehicles in the roundabout scenario. This case is also extracted from the INTERACTION-DR_USA_Roundabout_FT-vehicle_tracks_30 file. In this case, vehicle #119 in the DR_USA_Roundabout_FT-vehicle_track_30 file is replaced by the motion planners utilized in the paper. Likewise, its driving path is set as the reference path for the planners. And the planners need to predict the future trajectories of surrounding

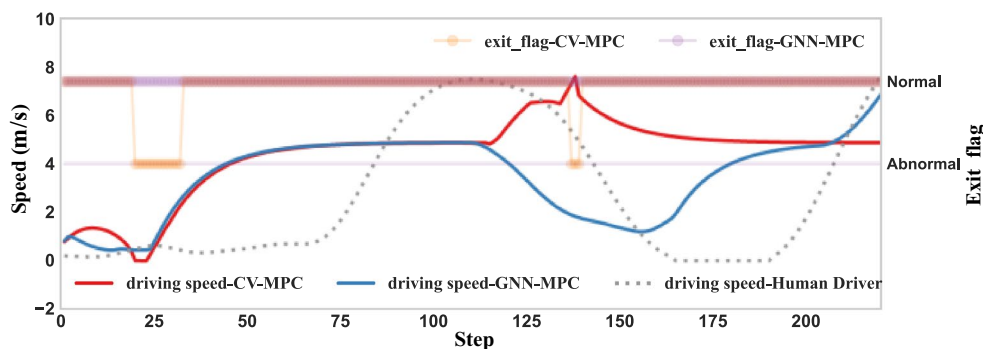


Figure 10 Comparison of driving velocity and safety performance in Case 2

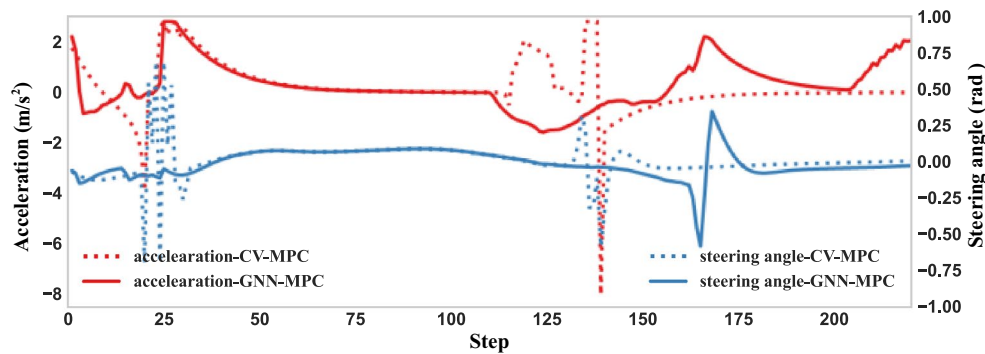


Figure 11 Comparison of acceleration and steering angle in Case 2

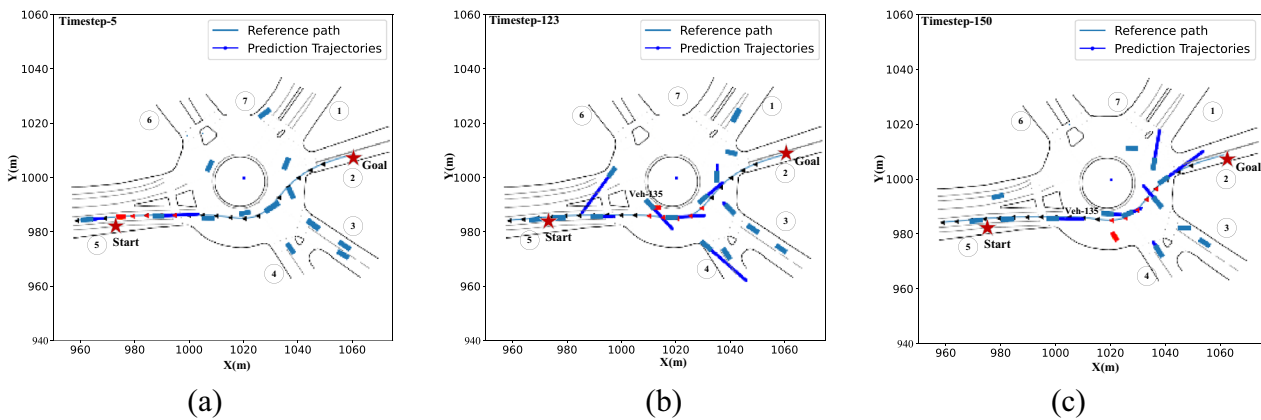


Figure 12 The motion planning results of CV-MPC for vehicle avoidance in Case 3, and (a–c) demonstrate three keyframes i.e., timestep-5, timestep-123, timestep-150

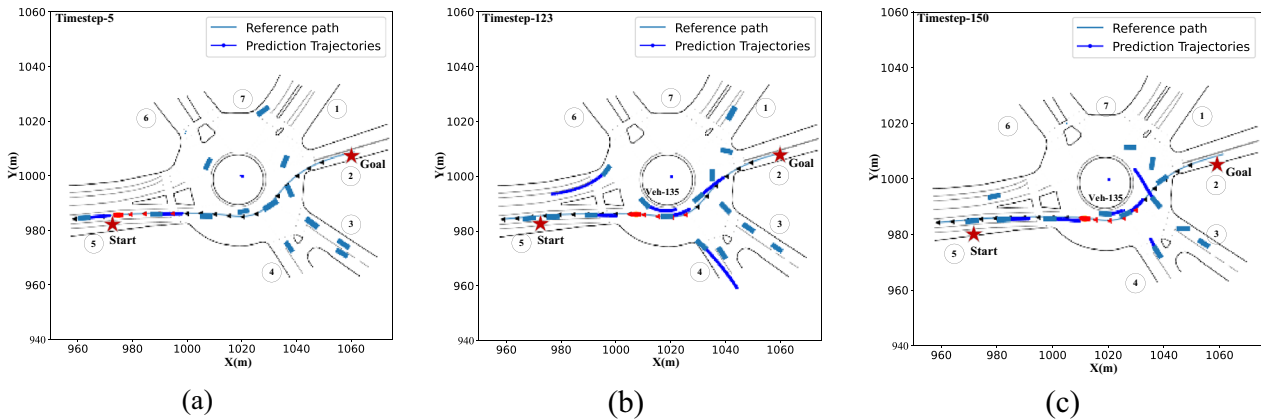


Figure 13 The motion planning results of GNN-MPC for vehicle avoidance in Case 3, and (a–c) demonstrate three keyframes i.e., timestep-5, timestep-123, timestep-150

road users to decide the acceleration and steering angle. The desired driving velocity is set as 8 m/s (i.e., $v_{ref} = 8$ m/s) in this case, and the distance that the ego vehicle can deviate from the reference path is set to 0.5 m (i.e., $W_{left}^{road} = W_{right}^{road} = 0.5$ m).

From Figure 12, it can be found that the CV-MPC fails in this case. The reason is that it predicts the future trajectory of vehicle #135 deviates significantly from the actual trajectory. The planned results of GNN-MPC are given in Figure 13, which illustrate that it can accurately

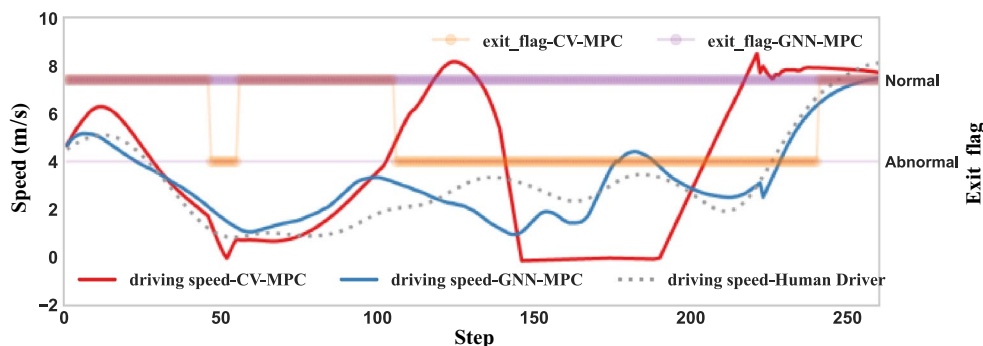


Figure 14 Comparison of driving velocity and safety performance in Case 3

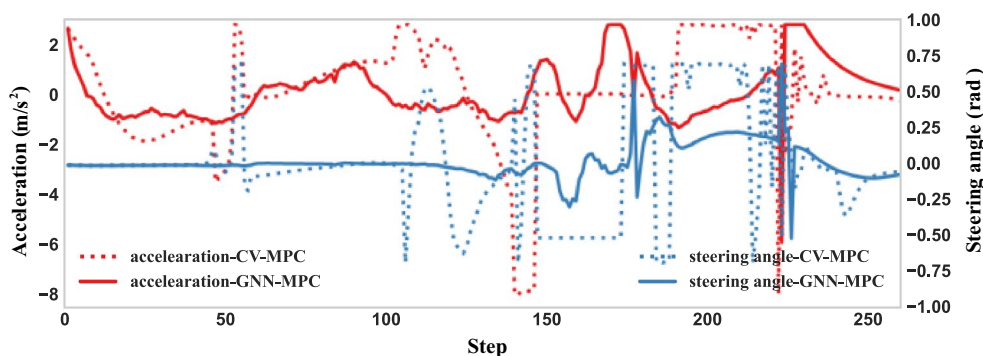


Figure 15 Comparison of acceleration and steering angle in Case 3

predict the future motion of key vehicle #135 and pass the roundabout safely. The safety metric, i.e., the `exit_flag` of the FORCESPRO solver and driving velocity is also shown in Figure 14. Finally, the control variables are given in Figure 15.

6 Conclusions

- (1) A trajectory prediction model based on the GNN was developed, which was utilized to forecast the future trajectories of nearby road users, such as pedestrians and vehicles.
- (2) A GNN-enabled motion planner based on MPC was established. Two driving scenarios, i.e., merging and roundabout scenarios were taken from the INTERACTION dataset to validate and verify the efficacy of the proposed motion planning approach.
- (3) The experiment results show that compared to the baseline technique, the proposed method can reduce risk and increase driving safety.

Acknowledgments
Not applicable.

Authors' Contributions

XT was in charge of the whole trial; WC wrote the manuscript; KY and SL assisted with experimental analyses. All authors read and approved the final manuscript.

Authors' Information

Wenbo Chu born in 1986, is currently a Research Fellow at *China Intelligent and Connected Vehicles (Beijing) Research Institute Co., Ltd., China*, which is also the *National Innovation Center of Intelligent and Connected Vehicles, China*. He received his B.S. degree majored in automotive engineering from *Tsinghua University, China*, in 2008, and his M.S. degree majored in automotive engineering from *RWTH-Aachen, German*, and Ph.D. degree majored in mechanical engineering from *Tsinghua University, China*, in 2014.

Kai Yang born in 1995, is currently pursuing a Ph.D. degree in *College of Mechanical and Vehicle Engineering, Chongqing University, China*. He received a B.E. degree in vehicle engineering from *Wuhan University of Technology, China*, in 2018. His research interests include automated vehicles, motion planning, and control.

Shen Li born in 1989, received the Ph.D. degree at *University of Wisconsin-Madison, USA*, in 2018. He is a research associate at *Tsinghua University, China*. His research interests include intelligent transportation systems (ITS), architecture design of CAVH system, vehicle-infrastructure cooperative planning and decision method, traffic data mining based on cellular data, and traffic operations and management.

Xiaolin Tang born in 1984, is currently a Professor at the *College of Mechanical and Vehicle Engineering, Chongqing University, China*. He received a Ph.D. degree in mechanical engineering from *Shanghai Jiao Tong University, China*, in 2015. His research interests include hybrid electric vehicles (HEVs), vehicle dynamics, energy management, and autonomous vehicle. He also serves as the Associate Editor of *IEEE Transactions on Vehicular Technology*, and *IEEE Transactions on Transportation Electrification*.

Funding

Supported by National Natural Science Foundation of China (Grant Nos. 52222215, 52072051), and Chongqing Municipal Natural Science Foundation of China (Grant No. CSTB2023NSCQ-JQX0003).

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Competing Interests

The authors declare no competing financial interests.

Received: 3 December 2022 Revised: 19 October 2023 Accepted: 22 October 2023

Published online: 10 January 2024

References

- X Tang, J Chen, K Yang, et al. Visual detection and deep reinforcement learning-based car following and energy management for hybrid electric vehicles. *IEEE Transactions on Transportation Electrification*, 2022, 8(2): 2501-2515.
- K Yang, X Tang, J Li, et al. Uncertainties in onboard algorithms for autonomous vehicles: Challenges, mitigation, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(9): 8963-8987.
- Z Zhou, L Ye, J Wang, et al. Hivt: Hierarchical vector transformer for multi-agent motion prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022: 8823-8833.
- X Tang, K Yang, H Wang, et al. Prediction-uncertainty-aware decision-making for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 2022, 7(4): 849-862.
- Y Huang, J Du, Z Yang, et al. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2022, 7(3): 652-674.
- S Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 23(2): 740-759.
- B Paden, M Čáp, S Z Yong, et al. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 2016, 1(1): 33-55.
- K Yang, X Tang, Y Qin, et al. Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust H-infinity state feedback control. *Chinese Journal of Mechanical Engineering*, 2021, 34: 74.
- N Kaempchen, B Schiele, K Dietmayer. Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2009, 10(4): 678-687.
- S Ammoun, F Nashashibi. Real time trajectory prediction for collision risk estimation between vehicles. *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, IEEE, 2009: 417-422.
- S Lefèvre, D Vasquez, C Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 2014, 1(1): 1-14.
- N Deo, M M Trivedi. Convolutional social pooling for vehicle trajectory prediction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018: 1468-1476.
- X Mo, Y Xing, C Lv. Graph and recurrent neural network-based vehicle trajectory prediction for highway driving. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021: 1934-1939.
- S Kumar, Y Gu, J Hoang, et al. Interaction-based trajectory prediction over a hybrid traffic graph. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021: 5530-5535.
- X Mo, Y Xing, C Lv. Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2020: 5057-5062.
- X Li, X Ying, M C Chuah. Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. arXiv preprint [arXiv: 1907.07792](https://arxiv.org/abs/1907.07792), 2019.
- F Diehl, T Brunner, M T Le, et al. Graph neural networks for modelling traffic participant interaction. *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019: 695-701.
- X Mo, Z Huang, Y Xing, et al. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(7): 9554-9567.
- L Xin, P Wang, C Y Chan, et al. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018: 1441-1446.
- J Li, S Liu, B Zhang, et al. RRT-A* motion planning algorithm for non-holonomic mobile robot. *2014 Proceedings of the SICE Annual Conference (SICE)*, IEEE, 2014: 1833-1838.
- J Y Hwang, J S Kim, S S Lim, et al. A fast path planning by path graph optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2003, 33(1): 121-129.
- W Schwarting, J Alonso-Mora, D Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018, 1: 187-210.
- H Wang, B Lu, J Li, et al. Risk assessment and mitigation in local path planning for autonomous vehicles with LSTM based predictive model. *IEEE Transactions on Automation Science and Engineering*, 2021, 19(4): 2738-2749.
- Y Chen, S Li, X Tang, et al. Interaction-aware decision making for autonomous vehicles. *IEEE Transactions on Transportation Electrification*, 2023, 9(3): 4704-4715.
- Y Rasekhipour, A Khajepour, S K Chen, et al. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2016, 18(5): 1255-1267.
- Y Huang, H Ding, Y Zhang, et al. A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach. *IEEE Transactions on Industrial Electronics*, 2019, 67(2): 1376-1386.
- M Zhu, X Wang, Y Wang. Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 2018, 97: 348-368.
- X Tang, B Huang, T Liu, et al. Highway decision-making and motion planning for autonomous driving via soft actor-critic. *IEEE Transactions on Vehicular Technology*, 2022, 71(5): 4706-4717.
- K Yang, X Tang, S Qiu, et al. Towards robust decision-making for autonomous driving on highway. *IEEE Transactions on Vehicular Technology*, 2023, 72(9): 11251-11263.
- S Bae, D Isele, A Nakhaei, et al. Lane-change in dense traffic with model predictive control and neural networks. *IEEE Transactions on Control Systems Technology*, 2023, 31(2): 646-659.
- Z Sheng, L Liu, S Xue, et al. A cooperation-aware lane change method for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(3): 3236-3251.
- P Gupta, D Isele, D Lee, et al. Interaction-aware trajectory planning for autonomous vehicles with analytic integration of neural networks into model predictive control. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023: 7794-7800.
- W Zhan, L Sun, D Wang, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint [arXiv:1910.03088](https://arxiv.org/abs/1910.03088), 2019.
- B Brito, A Agarwal, J Alonso-Mora. Learning interaction-aware guidance for trajectory optimization in dense traffic scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(10): 18808-18821.
- W Schwarting, J Alonso-Mora, L Pauli, et al. Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017: 1928-1935.
- K Yang, B Li, W Shao, et al. Prediction failure risk-aware decision-making for autonomous vehicles on signalized intersections. *IEEE Transactions on Intelligent Transportation Systems*, 2023. <https://doi.org/10.1109/ITIS.2023.3288507>.
- A Zanelli, A Domahidi, J Jerez, et al. FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, 2020, 93(1): 13-29.