

Sheng, Z., Liu, L., Xue, S., Zhao, D., Jiang, M. and Li, D. (2023) A cooperation-aware lane change method for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(3), pp. 3236-3251.



Copyright © 2022 IEEE. Reproduced under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

For the purpose of open access, the author(s) has applied a Creative Commons Attribution license to any Accepted Manuscript version arising.

<https://eprints.gla.ac.uk/285950/>

Deposited on: 22 November 2022

A Cooperation-Aware Lane Change Method for Automated Vehicles

Zihao Sheng, Lin Liu, Shibeixue, *Senior Member, IEEE*, Dezhong Zhao, *Senior Member, IEEE*,
Min Jiang, Dewei Li

Abstract—Lane change for automated vehicles (AVs) is an important but challenging task in complex dynamic traffic environments. Due to difficulties in guarantee safety as well as a high efficiency, AVs are inclined to choose relatively conservative strategies for lane change. To avoid the conservatism, this paper presents a cooperation-aware lane change method utilizing interactions between vehicles. We first propose an interactive trajectory prediction method to explore possible cooperations between an AV and the others. Further, an evaluation is designed to make a decision on lane change, in which safety, efficiency and comfort are taken into consideration. Thereafter, we propose a motion planning algorithm based on model predictive control (MPC), which incorporates AV's decision and surrounding vehicles' interactive behaviors into constraints so as to avoid collisions during lane change. Quantitative testing results show that compared with the methods without an interactive prediction, our method enhances driving efficiencies of the AV and other vehicles by 14.8% and 2.6% respectively, which indicates that a proper utilization of vehicle interactions can effectively reduce the conservatism of the AV and promote the cooperation between the AV and others.

Index Terms—Decision making, motion planning, collision avoidance, automated vehicles.

I. INTRODUCTION

AUTOMATED driving technologies are believed to have potentials to significantly reduce traffic accidents, improve travel efficiencies, and enhance comforts of drivers and passengers [1], [2]. To this end, AVs should have abilities to complete different tasks, such as cruise, lane change, etc. Lane change therein is regarded as one of the most dangerous

This work was supported in part by the National Natural Science Foundation of China under Grant 62273226, and Grant 61873162; in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT2021B24; in part by the Scientific Research Funding of Shanghai Dianji University, under Grant B1-0288-21-007-01-023; and in part by the Engineering and Physical Sciences Research Council of the U.K., through the EPSRC Innovation Fellowship, under Grant EP/S001956/1. (*Corresponding author: Shibeixue.*)

Zihao Sheng, Shibeixue and Dewei Li are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, P. R. China, also with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, P. R. China, and also with the Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai 200240, P. R. China (e-mail: zihaozheng@sjtu.edu.cn; shbxue@sjtu.edu.cn; dwli@sjtu.edu.cn).

Lin Liu is with the School of Electronic Information Engineering, Shanghai Dianji University, Shanghai 201306, P. R. China (e-mail: liul@sdju.edu.cn).

Dezhong Zhao is with the James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom (e-mail: Dezhong.Zhao@glasgow.ac.uk).

Min Jiang is with the School of Electronics and Information Engineering, Soochow University, Suzhou 215006, P. R. China (e-mail: jiang-min08@suda.edu.cn)

tasks, due to interactions with surrounding vehicles, and causes many traffic accidents. However, lane change would benefit to greatly save travelling times and ensure safety, when it is performed at a right time and in a proper way. In order to enhance the performance of automated driving, many studies focus on decision making and motion planning associated with lane change.

In the aspect of decision making on lane change, rule-based approaches were widely used in previous works since they are clear in logic and with low computational costs. For example, Wei *et al.* [3] proposed a decision making framework to determine a target lane and an acceleration, where a distance-keeping model predicts other vehicles' trajectories and a cost function was designed for selection of the best decision. Moridpour *et al.* [4] proposed a fuzzy logic model for decision in lane change, in which a fuzzy rule set was defined to explain decision making processes. However, rule-based methods lack flexibility and only handle limited scenarios because of the difficulty in the logic design for complex or unknown scenarios. In addition, it is difficult for rule-based methods to handle uncertainties. To outperform it, Markov decision process (MDP) [5], [6] has been studied to provide robust lane change decisions. However, it is generally difficult to solve MDP in real time due to its high computational complexity [7].

In recent years, with the development of machine learning, learning-based methods provide a new avenue for lane change decisions. Machine learning models not only can deal with uncertainties due to their probabilistic reasoning, but also can learn nonlinear mapping relations from real-world data to decisions [8]–[10]. In this way, learning-based methods can make human-like decisions to improve the robustness of decisions. For example, Hou *et al.* [8] developed a lane-change-assisted framework, which combines a decision tree and a Bayes classifier to predict whether other drivers decide to merge. Gu *et al.* [11] used a deep autoencoder network to identify driving behaviors and a XGBoost model to decide whether to change lanes. Other learning-based approaches were also proposed for lane change decisions including support vector machine [12], [13], deep neural network [14], [15] and reinforcement learning [16]–[18]. However, due to the lack of interpretability on machine learning algorithms, it is difficult to find the reason of a failure decision under some scenarios.

On the other hand, to complete lane change, an AV with a decision needs to plan a proper trajectory that satisfies specific constraints, e.g., vehicle dynamics, collision avoidance, and available times. In the existing works, motion planning

methods can be generally classified into four categories: graph search, interpolating curve, sampling, and numerical optimization [19]. The commonly used graph-search-based methods include D^* algorithm [20], field D^* algorithm [21], Dijkstra algorithm [22], and hybrid A^* algorithm [23]. However, these methods rarely consider constraints of vehicle kinematics, so the generated paths are usually difficult to track. In addition, interpolating-curve-based approaches are usually efficient in computation, such as polynomial curves [24], [25], Bezier curves [26], [27], Dubins curves [28], and Clothoids curves [29]. Nevertheless, these curves are sequences of positions without considering the variation of positions over time, so they cannot guarantee to avoid collisions. Moreover, sampling-based methods sample a configuration space and select samples satisfying predefined requirements as results. Many sampling-based methods were developed based on rapidly-exploring random trees (RRT), since it can deal with general dynamic models [30]–[32]. However, their results would be unstable. Finally, numerical-optimization-based approaches obtain an optimal trajectory by minimizing an objective function subject to a set of constraints [33], [34]. This kind of method can comprehensively incorporate various objectives and constraints in trajectory planning.

Although the above approaches have made great progresses in improving AVs’ capacities to execute lane change, there are still limitations. Firstly, these methods take the decision making and trajectory planning for lane change as two separate problems. However, they are actually highly interdependent, so the separation of them would lead to inconsistency between each other. For instance, if the mechanical and physical limits of vehicle motions are not considered by decision making, the motion planning module may fail to find a feasible solution. Secondly, potential cooperations between an AV and other vehicles have not been fully explored, resulting in conservative behaviors of AVs in some complex scenarios. As a specific example, in dense traffic, an AV would not change lanes if it cannot cooperate with others.

Therefore, to overcome the above limitations and improve the safety and efficiency of AVs, this paper presents a cooperation-aware method for determining a proper time instant and a maneuver in lane change, which takes interactions among vehicles into account. Instead of separately designing the modules of decision making and motion planning, we consider these two modules in a unified framework. In detail, at the higher level, a cost-function-based and deep-learning-based algorithm is proposed for solving the decision-making problems. The constraints of trajectory planning imposed by the motion capabilities of the ego vehicle, current road conditions and traffic environment (e.g., acceleration limits, road curvature, number of lanes, speed limits, etc.) are considered in advance during the decision making to customize the decision set. By doing so, the consistency between the decisions generated at the higher level and the trajectory planning in the lower level can be well guaranteed. In this way, our method can also be easily extended to various traffic scenarios, such as highways, overpasses, urban intersections, etc. In addition, we propose an interactive trajectory prediction method to predict surrounding vehicles’ reactions to each decision, which allows

to explore possible cooperations. Based on these predictions, an optimal decision is selected by considering safeties, efficiencies and comforts. Subsequently, an MPC-based motion planning algorithm is proposed to solve a trajectory for lane change. Here, we incorporate the prediction of interactive trajectories into constraints to effectively use vehicular interactions to plan a safe trajectory. The contributions of this paper are two folds.

(1) Different from existing works, we propose a complete lane change method, which considers the correlation between decision making and trajectory planning to enhance safeties and efficiencies for lane change and prevent conflicts between them. In this way, this method avoids the situations that the planning module fails to generate a trajectory due to an improper decision, and thus guarantees a smooth driving.

(2) Compared with existing methods, we propose an interactive trajectory prediction method, which helps an AV make proactive decisions and avoid being shortsighted. The comparison with other methods demonstrates that it improves driving efficiencies of the AV and others by 14.8% and 2.6%, demonstrating that the AV can use interactions to cooperate with others, and thus to achieve an efficient driving.

The rest of this paper is organized as follows. The problem description is given in Section II. Our system structure is described in Section III. We present a detailed description of the lane change decision module in Section IV. The lane change trajectory planning module is introduced in Section V. Experimental results and analysis are shown in Section VI. We analyze the effects of different hyperparameters and the necessity of considering module interdependence in Section VII. Finally, Section VIII concludes this paper.

II. PROBLEM DESCRIPTION OF LANE CHANGE

Lane change is a key maneuver for automated driving, which can help an AV secure a higher driving efficiency and safety. In this paper, we consider that only one vehicle is controlled by our proposed lane change method, referred as “ego vehicle”, and all vehicles are not connected through vehicle-to-vehicle or vehicle-to-infrastructure. All vehicles are passenger cars with similar shapes and motion patterns; motorcycles, buses or heavy trucks are not considered. Since a lane change is not recommended near intersections or traffic signals, we assume that the road remains straight along a lane change.

Lane change is executed in the form of a trajectory, which is a function of time mapping time to the state of a vehicle. Hence, we define a trajectory of an ego vehicle as $\tau : \mathbb{R} \rightarrow \mathbb{R}^C$. For the convenience of trajectory planning, the resulting state is denoted as,

$$\tau(t) = [x(t), y(t), \psi(t), v(t)]^\top, \quad (1)$$

where x and y denote longitudinal and lateral positions of a vehicle, respectively, ψ represents the heading angle, and v denotes the velocity. Here, note that these four components are studied for lane change and thus we have $C = 4$. When a vehicle runs on a road, the drivable areas impose boundaries on x and y . Besides, v and ψ are limited by the motion capacity of a vehicle.

When the ego vehicle runs on a road with at least two lanes, we assume that it observes perfect trajectories of N vehicles around it without any noise or delay. Since the collected data is processed in a digital way, trajectories are discretized with a sampling period Δt . The discrete trajectory is denoted as $\tau[k] = \tau(k\Delta t)$. With an observation horizon T_o , the observed trajectory of the n th vehicle is denoted as,

$$\tau_n^o = \tau_n[k] = [x_n[k], y_n[k], \psi_n[k], v_n[k]]^\top, 1 \leq k \leq O, \quad (2)$$

where the subscript n indicates the trajectory of the n th vehicle, $O = T_o/\Delta t$, $\tau_n[k]$ is the state of the n th vehicle at the time step k . Similarly, the trajectory of the ego vehicle over T_o is denoted as,

$$\tau_{ego}^o = \tau_{ego}[k] = [x_{ego}[k], y_{ego}[k], \psi_{ego}[k], v_{ego}[k]]^\top, \quad 1 \leq k \leq O. \quad (3)$$

After the observation is obtained, the ego vehicle selects an optimal decision on lane change from a decision set \mathcal{D} . The decision set \mathcal{D} consists of B candidate decisions, which is denoted as $\mathcal{D} = \{\tau_i^d \mid \forall i \in \{1, \dots, B\}\}$. Each decision corresponds to a reference trajectory, denoted as,

$$\tau_b^d = \tau_b[k] = [x_b[k], y_b[k], \psi_b[k], v_b[k]]^\top, O+1 \leq k \leq O+D, \quad (4)$$

with a duration T_d , where the subscript b denotes the index of a decision, $D = T_d/\Delta t$, $\tau_b[k]$ is the state of the b th decision at the time step k . The optimal decision on lane change is denoted as τ_{opt}^d . Based on τ_{opt}^d , the ego vehicle adjusts its acceleration and steering angle to plan a trajectory over a prediction horizon T_p , which is represented as,

$$\tau_{ego}^p = \tau_{ego}[k], O+1 \leq k \leq O+P, \quad (5)$$

where $P = T_p/\Delta t$.

Thus, the problem of lane change is summarized as follows. Given observed trajectories of the ego vehicle and its surrounding vehicles, i.e., τ_{ego}^o and $\{\tau_1^o, \dots, \tau_n^o, \dots, \tau_N^o\}$, the ego vehicle selects an optimal decision τ_{opt}^d from a decision set \mathcal{D} , and finally plans a trajectory τ_{ego}^p by adjusting its acceleration and steering angle.

III. STRUCTURE OF PROPOSED COOPERATION-AWARE LANE CHANGE METHOD

As mentioned before, most studies on lane change treat decision making and trajectory planning as two separate problems. The decision making module provides high-level orders, such as accelerating/decelerating, left/right lane change, to low-level modules. The trajectory planning module receives these orders and then solves a feasible trajectory. Although decision making and trajectory planning are hierarchical in term of modules, they are actually highly coupled for successful executions [35], [36]. Therefore, the separate design of the decision making and trajectory planning causes that an improper decision from the decision making module is sent to the trajectory planning module, which might fail to generate a feasible trajectory.

To solve the above problem, we propose a cooperation-aware lane change method, which considers the correlation

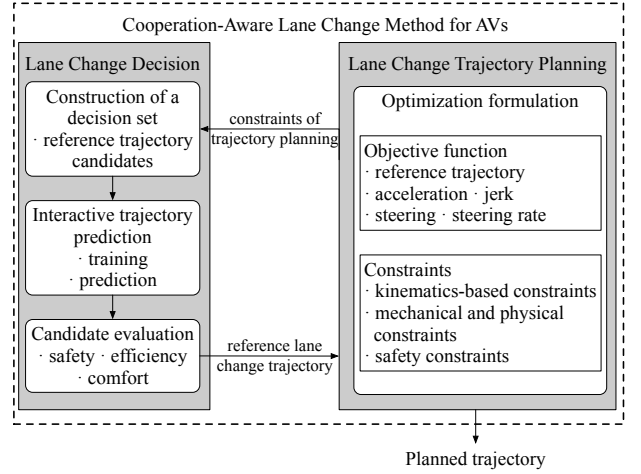


Fig. 1. System architecture of the proposed cooperation-aware lane change method for automated vehicles.

between the decision making and trajectory planning. The structure of the proposed method is shown in Fig. 1. We can see that the two modules are interdependent. The planning module first sends to the decision module constraints of trajectory planning caused by the motion capabilities of the ego vehicle, current road conditions and traffic environment (e.g., acceleration limits, road curvature, number of lanes, speed limits, etc.), so that the feasibility of trajectory planning can be considered in decision making. Then, the decision module provides a reference trajectory for the planning module, which uses it to generate a planned trajectory.

The main aim of the decision module is to determine an appropriate time and a proper speed to change lanes while considering the limits of vehicle motions. We first generate various reference trajectory candidates, which consider the feasibility of trajectory planning, to construct a decision set \mathcal{D} . An interactive trajectory prediction method is then proposed to explore which reference trajectory candidate would potentially promote cooperations between the ego vehicle and surrounding vehicles. Finally, an optimal decision τ_{opt}^d is selected from \mathcal{D} based on the interactive trajectory prediction while considering safety, driving efficiency and comfort.

The role of the trajectory planning module is to generate a dynamically feasible, collision-free and smooth trajectory based on τ_{opt}^d . The optimal decision τ_{opt}^d poses constraints for the trajectory planning. We formulate the trajectory planning as an optimization problem, which takes model predictive control (MPC) as the basis, since MPC can comprehensively consider the objective function and various constraints, including vehicle dynamics, safety, smooth control actions, ect. By solving this optimization problem, a qualified trajectory is planned.

IV. LANE CHANGE DECISION BASED ON INTERACTIVE TRAJECTORY PREDICTION

In this section, we introduce details about the decision module for lane change in the left side of Fig. 1, including the construction of decision set, interactive trajectory prediction, and candidate evaluation.

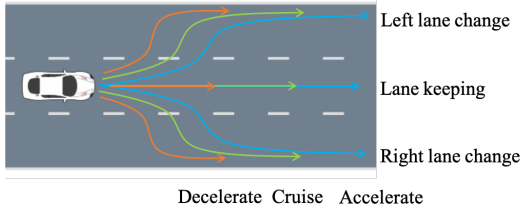


Fig. 2. Predefined reference trajectory candidates which combine three longitudinal options and three lateral actions.

A. Construction of a decision set

The decision set \mathcal{D} consists of various candidate decisions. Each decision τ_b^d is a reference trajectory, which is an initialization for the subsequent interactive prediction and trajectory planning. For the computational convenience, we set the duration of each reference trajectory T_d to be equal to T_p . In this paper, we limit our discussion on combining three lateral actions (left lane change, right lane change and staying in the current lane) and three longitudinal options (accelerating, decelerating, and keeping in the current speed range) in Fig. 2 to construct \mathcal{D} . It is worth noting that \mathcal{D} can be extended to contain more decisions according to a practical demand. For instance, in a dense traffic, \mathcal{D} can consist of decisions with more gentle lane changes (i.e., slower lateral speeds).

In order to generate each τ_b^d , we first calculate the lateral and the longitudinal accelerations. Inspired by the work in [37], we use the PNTLA method to generate a continuous-time lateral reference trajectory. This method can calculate the smooth trajectory with curvature continuity based on the maximum lateral acceleration $a_{y \max}$ and the maximum lateral jerk $\Delta a_{y \max}$. As shown in Fig. 3, the PNTLA method assumes that when vehicles change lanes, the lateral acceleration varies linearly, and the acceleration curve is composed of two isosceles trapezoids with the same size but opposite directions. Hence, the lateral jerk Δa_y is expressed as,

$$\Delta a_y(t) = \begin{cases} \Delta a_{y \max} & 0 < t < t_1 \\ 0 & t_1 \leq t < t_2 \\ -\Delta a_{y \max} & t_2 \leq t < t_3 \\ 0 & t_4 \leq t < t_4 \\ \Delta a_{y \max} & t_4 \leq t < t_5 \end{cases}, \quad (6)$$

where t_1 to t_5 are five key timestamps. Based on the properties of isosceles trapezoids, they are calculated as [37],

$$\begin{cases} t_1 = \frac{a_{y \max}}{\Delta a_{y \max}}, \\ t_2 = -\frac{t_1}{2} + \frac{\sqrt{t_1^2 + 4d_w/a_{y \max}}}{2}, \\ t_3 = 2t_1 + t_2, \\ t_4 = t_1 + 2t_2, \\ t_5 = 2t_1 + 2t_2, \end{cases} \quad (7)$$

where d_w denotes the lateral displacement during a lane change. With the timestamps t_1 to t_5 , the lateral acceleration curve can be determined.

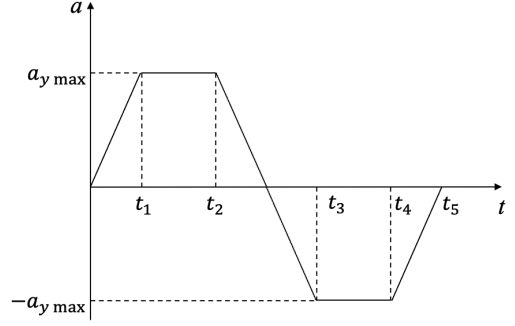


Fig. 3. Curve of lateral acceleration generated by the positive and negative trapezoidal lateral acceleration method.

The longitudinal reference trajectory is generated in a similar manner. The longitudinal acceleration curve is described as an isosceles trapezoid which guarantees the smoothness and curvature continuity of a generated trajectory. The longitudinal jerk Δa_x is expressed as,

$$\Delta a_x(t) = \begin{cases} \Delta a_{x \max} & 0 < t < t_6 \\ 0 & t_6 \leq t < t_7 \\ -\Delta a_{x \max} & t_7 \leq t < t_8 \end{cases}, \quad (8)$$

where $\Delta a_{x \max}$ denotes a maximum longitudinal jerk, t_6 to t_8 are three key timestamps. Based on the properties of isosceles trapezoids, t_6 to t_8 are calculated as,

$$\begin{cases} t_6 = \frac{a_{x \max}}{\Delta a_{x \max}}, \\ t_7 = \frac{|v_{x1} - v_{x0}|}{a_{x \max}}, \\ t_8 = t_6 + t_7, \end{cases} \quad (9)$$

where $a_{x \max}$ denotes a maximum longitudinal acceleration, v_{x0} is the current speed, v_{x1} is the desired speed. Similarly, with the timestamps t_6 to t_8 , the longitudinal acceleration curve is established.

After obtaining the lateral and longitudinal acceleration curves, we can calculate a continuous-time velocity profile by integrating the acceleration curves, and then calculate a continuous-time position profile by integrating the acceleration curves twice. Since the reference trajectory does not consider the vehicle dynamics, the heading angle is set to be equal to the angle between the direction of velocity and the longitudinal axis. Therefore, we obtain a continuous-time trajectory, which should be discretized with Δt to obtain each τ_b^d .

It is worth noting that to acceleration-dependent trajectory cannot be directly used to control the lateral motions. We need to convert acceleration into a controllable variable. To this end, we first calculate the specific information (e.g., speeds, heading angles, etc.) of each point on the acceleration-dependent trajectory. Then, the planning module can use a more complex model to describe vehicle motions more precisely. In this model, the lateral motions are controlled by steering angles, which is achieved by adjusting the steering wheel directly.

B. Interactive trajectory prediction of surrounding vehicles

To select a $\tau_b^d \in \mathcal{D}$ as an optimal decision τ_{opt}^d for the ego vehicle, it is crucial to predict how surrounding vehicles will

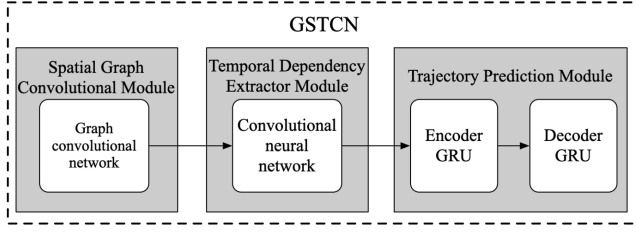


Fig. 4. Architecture of GSTCN [38].

react to each τ_b^d . However, due to the strong spatial-temporal dependencies among vehicles, predicting their trajectories is challenging. In this paper, we propose an interactive trajectory prediction method based on the graph-based spatial-temporal convolutional network (GSTCN) [38] to tackle this task, since this network can effectively capture the spatial-temporal dependencies among vehicles and then predict their future trajectories simultaneously. As shown in Fig. 4, GSTCN is composed of three modules: a spatial graph convolutional module, a temporal dependency extractor module, and a trajectory prediction module. The first module uses a graph convolution network to extract the spatial dependencies between vehicles. The following temporal dependency extractor module learns the temporal dependencies. The trajectory prediction module consists of an encoder and a decoder, where both of them are composed of the gated recurrent unit (GRU) networks to deal with the trajectory sequence generation. Note that GSTCN takes uncertainties of traffic into account as it directly learns from raw data with noise and measurement errors and predicts the probability distributions over future trajectories. In this way, it can handle the uncertainty in raw data and describe stochastic behaviors of human drivers. More details of the GSTCN can be found in [38]. Here, the proposed interactive trajectory prediction method will be introduced in detail from two stages: training step and prediction step.

1) *Training step*: GSTCN takes the observed trajectories of all vehicles over an observation horizon T_o as inputs and predicts their future coordinates simultaneously. GSTCN assumes future positions of vehicles are random variables satisfying bi-variable Gaussian distributions, so this network is trained by minimizing a negative log-likelihood loss. The training step is done offline.

2) *Prediction step*: After the training is done, we use GSTCN to predict surrounding vehicles' trajectories for each τ_b^d . For a new observation of the positions of surrounding vehicles at the current moment, GSTCN first predicts their positions at the next time step. Then, we use this prediction to update their current positions. Similarly, the current position of the ego vehicle is updated by τ_b^d . This process will be repeated until τ_b^d is used up. In this way, we obtain the future trajectory τ_n^f of each surrounding vehicle to each τ_b^d over a prediction horizon T_p .

To formalize, the proposed interactive trajectory prediction method is denoted as a function ϕ that maps τ_b^d , τ_{ego}^o , and $\{\tau_1^o, \dots, \tau_n^o, \dots, \tau_N^o\}$ to predicted interactive trajectories $\{\tau_{1,b}^f, \dots, \tau_{n,b}^f, \dots, \tau_{N,b}^f\}$.

C. Candidate evaluation method

After obtaining the predicted interactive trajectories $\{\tau_{1,b}^f, \dots, \tau_{n,b}^f, \dots, \tau_{N,b}^f\}$ under every $\tau_b^d \in \mathcal{D}$, we should determine which τ_b^d is the most recommended in terms of safety, efficiency and comfort [35], [39]. We evaluate each τ_b^d by calculating their costs. The cost function consists of three components and is calculated as

$$J_d(\tau_b^d) = \lambda_s J_s + \lambda_e J_e + \lambda_c J_c, \quad \tau_b^d \in \mathcal{D}, \quad (10)$$

where J_s , J_e , and J_c are the costs of safety, efficiency and comfort, respectively. λ_s , λ_e , and λ_c represent the corresponding weights, which can be customized by the user.

The safety cost J_s is related to the collision risks between the ego vehicle and nearby vehicles, which consists of risks in both lateral and longitudinal directions. Therefore, the safety cost J_s is represented as,

$$J_s = J_{s-lon} + \eta^2 J_{s-lat}, \quad (11)$$

where J_{s-lon} and J_{s-lat} represent the safety costs in longitudinal and lateral directions, respectively. $\eta = 1$ if the ego vehicle changes to other lanes, otherwise $\eta = 0$, which guarantees that if the τ_b^d does not relate to lane change, the lateral risks can be ignored.

The longitudinal safety cost J_{s-lon} relates to the collision risks between the ego vehicle and the preceding and following vehicles in the same lane. In this paper, we use the time-to-collision (TTC) [40] to characterize the collision risks, which is calculated based on the relative distances and velocities against the ego vehicle and the preceding and following vehicles in the same lane. The smaller TTC is, the larger the collision risk is. Thus, we define J_{s-lon} as,

$$J_{s-lon} = \kappa_{s-lon} \sum_{k=O+1}^{O+P} \left[\sigma\left(\frac{\Delta v_{pv-lon}[k]}{\Delta s_{pv-lon}[k]}\right) + \sigma\left(\frac{\Delta v_{fv-lon}[k]}{\Delta s_{fv-lon}[k]}\right) \right], \quad (12)$$

with

$$\begin{cases} \Delta v_{pv-lon}[k] = v_{ego}[k] - v_{pv-lon}[k], \\ \Delta v_{fv-lon}[k] = v_{ego}[k] - v_{fv-lon}[k], \\ \Delta s_{pv-lon}[k] = \|s_{ego}[k] - s_{pv-lon}[k]\|_2, \\ \Delta s_{fv-lon}[k] = \|s_{ego}[k] - s_{fv-lon}[k]\|_2, \end{cases} \quad (13)$$

where $v_{ego}[k]$ obtained from τ_b^d is the velocity of the ego vehicle at the time step k . The subscripts $pv-lon$ and $fv-lon$ represent the indices of the nearest preceding and following vehicles in the same lane, respectively. v_{pv-lon} is the velocity of the nearest preceding vehicle in the same lane at the time step k , which is obtained from $\{\tau_{1,b}^f, \dots, \tau_{n,b}^f, \dots, \tau_{N,b}^f\}$. $s = [x, y]^T$ denotes the position of a vehicle, and $\|\cdot\|_2$ is the l_2 norm. Hence, $\Delta v_{pv-lon}[k]$ and $\Delta s_{pv-lon}[k]$ denote the relative velocity and distance between the ego vehicle and the nearest preceding vehicle in the same lane at the time step k . Δv_{fv-lon} and Δs_{fv-lon} denote the relative velocity and distance between the ego vehicle and the nearest following vehicle in the same lane. κ_{s-lat} is the regularization coefficient of the longitudinal safety cost, which is determined by practical demands. $\sigma(\cdot) = \min(\cdot, 0)$, which guarantees that

two vehicles moving away from each other will not contribute a safety cost.

Similarly, the lateral safety cost J_{s-lat} is calculated based on the relative distances and velocities against the ego vehicle and the preceding and following vehicles in the target lane, which is defined as,

$$J_{s-lat} = \kappa_{s-lat} \sum_{k=O+1}^{O+P} \left[\sigma \left(\frac{\Delta v_{pv-lat}[k]}{\Delta s_{pv-lat}[k]} \right) + \sigma \left(\frac{\Delta v_{fv-lat}[k]}{\Delta s_{fv-lat}[k]} \right) \right], \quad (14)$$

where κ_{s-lat} is the regularization coefficient of the lateral safety cost. The subscripts $pv-lat$ and $fv-lat$ represent the indices of the nearest preceding and following vehicles in the target lane, respectively.

Moreover, the efficiency cost J_e is related to the velocities of the ego vehicle and surrounding vehicles. The closer their speeds are to the desired speed, the smaller the efficiency cost is. Hence, J_e is defined as,

$$J_e = \kappa_e \sum_{k=O+1}^{O+P} (v_{ego}[k] - v_{des})^2 + \frac{\kappa_e}{N} \sum_{n=1}^N \sum_{k=O+1}^{O+P} (v_n[k] - v_{des})^2, \quad (15)$$

where κ_e is the regularization coefficient of the efficiency cost, v_{des} is the desirable velocity, v_n and v_{ego} are the velocities of the n th nearby vehicle and the ego vehicle, respectively. Since we want to make decisions that can help the ego vehicle to cooperate with others, the efficiency cost indicates that both the ego vehicle and surrounding vehicles want to reach the desirable speeds. Note that v_{des} is defined based on the traffic density. Without loss of generality, we use an inverse proportional relation to describe the relationship between traffic density and desired velocity, i.e., the higher the density, the lower the desired velocity.

In addition, the comfort cost J_c aims to make the ego vehicle travel with a relatively smooth trajectory. Some existing works have focused on the relationship between jerk and comfort, where they found that jerk can cause psychological and physical discomforts, such as fear, dizziness and nausea [41], [42]. Thus, J_c is related to the jerk in both lateral and longitudinal directions. Besides, many existing trajectory planning methods [34], [35], [43], [44] adopt this idea to achieve better comfort. We define J_c as,

$$J_c = \sum_{k=O+1}^{O+P} [\kappa_{c-lon} (\Delta a_{ego-lon}[k])^2 + \kappa_{c-lat} (\Delta a_{ego-lat}[k])^2], \quad (16)$$

where κ_{c-lon} and κ_{c-lat} are regularization coefficients of the comfort cost. $\Delta a_{ego-lon}$ and $\Delta a_{ego-lat}$ denote the longitudinal and lateral jerks of the ego vehicle, which can be obtained by calculating the derivative of acceleration with respect to time from τ_b^d .

After the cost function is defined, we select a $\tau_b^d \in \mathcal{D}$ with the minimal cost J_d as the optimal decision τ_{opt}^d , which makes a balance among safety, efficiency, and comfort.

Note that in the process of generating the reference trajectory candidates, we utilize the lateral displacement, the maximum lateral acceleration, the maximum lateral jerk, the

maximum longitudinal acceleration, and the maximum longitudinal jerk. Therefore, the feasibility of a motion planning has been considered during the phase of the lane change decision.

V. MODEL-PREDICTIVE-CONTROL-BASED TRAJECTORY PLANNING FOR LANE CHANGE

In this section, based on the above optimal reference trajectory τ_{opt}^d and the corresponding interactive trajectory prediction $\{\tau_{1,opt}^f, \dots, \tau_{n,opt}^f, \dots, \tau_{N,opt}^f\}$, we introduce the trajectory planning for a lane change. We formulate the trajectory planning as a model predictive control (MPC)-based optimization problem. MPC can effectively handle optimization problems with constraints while respecting vehicle dynamics. Next, we will introduce a predictive model, constraints, an objective function, and the Ipopt-based approach for trajectory planning.

A. Kinematics-based predictive model

To realize the MPC-based trajectory planning for the ego vehicle, we first need a predictive model to predict vehicle motions, such as kinematic and dynamic models [45], car-following models [46] and neural networks [47]. The more accurately the model describes the vehicle kinematics and dynamics, the more accurately the planned trajectory will be tracked by a control system. However, due to the limited computing resources and high requirements for computing time in driving environments, the predictive model should also have the advantages of easy implementation and low computation complexity. To this end, we use the bicycle model in [45] to describe the nonlinear vehicle kinematics. With the Euler discretization, this vehicle kinematic model can be defined as,

$$\begin{cases} x[k+1] = x[k] + \Delta t \times v[k] \cos(\psi[k] + \beta[k]), \\ y[k+1] = y[k] + \Delta t \times v[k] \sin(\psi[k] + \beta[k]), \\ \psi[k+1] = \psi[k] + \Delta t \times \frac{v[k]}{l_r} \sin(\beta[k]), \\ v[k+1] = v[k] + \Delta t \times a[k], \\ \beta[k] = \tan^{-1} \left(\frac{l_r}{l_r + l_f} \tan(\delta[k]) \right), \end{cases} \quad (17)$$

where β denotes the angle between the speed and the heading, l_f and l_r denote the length from the gravity center of a vehicle to the front axle and the rear axle, respectively, a is the acceleration, and δ denotes the steering angle. Thus, we obtain a predictive model for MPC. As mentioned in Section II, the acceleration and the steering angle are taken as control actions, and thus we define them as $\mathbf{u}[k] = [a[k], \delta[k]]^T$.

B. Mechanical and physical constraints

Due to the mechanical and physical limitations in velocity, heading angle, and position, we need to take the boundaries of these variables as constraints in the process of motion planning to achieve a feasible solution. Specifically, they are represented as,

$$\begin{cases} \tau_{\min} \leq \tau[k] \leq \tau_{\max}, \\ \mathbf{u}_{\min} \leq \mathbf{u}[k] \leq \mathbf{u}_{\max}, \\ \Delta \mathbf{u}_{\min} \leq \mathbf{u}[k] - \mathbf{u}[k-1] \leq \Delta \mathbf{u}_{\max}, \end{cases} \quad (18)$$

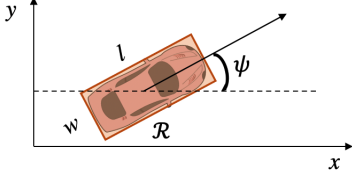


Fig. 5. The road area occupied by a vehicle is represented by a rectangle rotated by the heading angle ψ in the two-dimensional plane.

where the first constraint considers the road boundaries and the available range of the heading angle and the velocity, i.e., τ_{\min} and τ_{\max} ; the second constraint ensures the acceleration and the steering angle do not exceed the vehicle's capacity from \mathbf{u}_{\min} to \mathbf{u}_{\max} ; the last constraint guarantees the smoothness and comfort of generated trajectories by restraining the jerks and steering angle rates to a range from $\Delta\mathbf{u}_{\min}$ to $\Delta\mathbf{u}_{\max}$.

C. Safety constraints

In addition to the above mechanical and physical constraints, we also consider the safety constraints. Since the vehicle-occupied road area imposes restrictions on the movements of other vehicles, it is essential to model a precise vehicle shape. Unlike previous works that model the shape of vehicles as a mass point or particles with radial gaps, we assume vehicles are rectangular which is closer to an actual circumstance. As shown in Fig. 5, the vehicle-occupied road region is represented by a rectangle \mathcal{R} , which is expressed as,

$$\mathcal{R} = \{\mathbf{p} = [p_1, p_2]^T \mid \mathbf{A} \cdot \mathbf{p} \leq \mathbf{b}\}. \quad (19)$$

As in [48], the matrix \mathbf{A} and the vector \mathbf{b} are expressed as,

$$\mathbf{A} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \\ -\cos \psi & -\sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix}, \quad (20)$$

$$\mathbf{b} = [l/2, w/2, l/2, w/2]^T + \mathbf{A} \cdot [x_0, y_0]^T,$$

with the length l and width w of a vehicle, and the current coordinate of the gravity center $[x_0, y_0]$.

With this vehicle shape model, the distance between two vehicles can be represented as,

$$\text{dist}(\mathcal{R}_1, \mathcal{R}_2) = \left\{ \min_{\mathbf{p}_1, \mathbf{p}_2} \|\mathbf{p}_1 - \mathbf{p}_2\|_2 \mid \mathbf{p}_1 \in \mathcal{R}_1, \mathbf{p}_2 \in \mathcal{R}_2 \right\}, \quad (21)$$

where \mathcal{R}_1 and \mathcal{R}_2 are the road regions occupied by two vehicles, respectively, and \mathbf{p}_1 and \mathbf{p}_2 denote any coordinates in the occupied road regions. Therefore, the safety constraints is represented as,

$$\text{dist}(\mathcal{R}_{ego}[k], \mathcal{R}_n[k]) > d_{\min}, \forall n \in \{1, \dots, N\}, \quad (22)$$

where $\mathcal{R}_{ego}[k]$ and $\mathcal{R}_n[k]$ denote the road regions occupied the ego vehicle and the n th nearby vehicle at the time step k , respectively, d_{\min} is the minimum safety distance between vehicles. With the interactive trajectory prediction of nearby vehicles $\{\tau_{1,opt}^f, \dots, \tau_{n,opt}^f, \dots, \tau_{N,opt}^f\}$, we can calculate their occupied regions in the planning time horizon.

In this way, vehicular interactions are considered during trajectory planning.

Note that the collision avoidance constraint (22) is non-differentiable, so we adopt the approach in [49] to convert this constraint to a smooth and differentiable constraint by using strong duality theory. Based on this method, the safety constraints (22) can be reformulated as,

$$\begin{cases} -\mathbf{b}[k]^T \lambda_n[k] - \mathbf{b}_n^T[k] \mu_n[k] \geq d_{\min} \\ \mathbf{A}[k]^T \lambda_n[k] + \rho_n[k] = 0 \\ \mathbf{A}_n^T[k] \mu_n[k] - \rho_n[k] = 0 \\ \rho_n^T[k] \rho_n[k] \leq 1 \\ \lambda_n[k] \geq 0 \\ \mu_n[k] \geq 0 \end{cases}, \forall n \in \{1, \dots, N\}, \quad (23)$$

where λ_n , μ_n and ρ_n are the dual variables. Interested readers are referred to Ref. [49] for more details. With this exact and smooth reformulation, standard nonlinear solvers can deal with the safety constraints.

D. Objective function and overall optimization formulation for planning

The objective is to minimize the deviation between the trajectory updated by the vehicle kinematic model and the reference trajectory associated with the optimal decision; meanwhile, smooth control actions are preferred for comfort. This objective function is defined as follows,

$$J_p = \sum_{i=0}^M \|\mathbf{Q}_\tau(\tau[k+i|k] - \tau_{opt}^d[k+i])\|_2^2 + \sum_{i=0}^{M-1} \|\mathbf{Q}_u \mathbf{u}[k+i]\|_2^2 + \sum_{i=1}^{M-1} \|\mathbf{Q}_{\Delta u} \Delta \mathbf{u}[k+i]\|_2^2, \quad (24)$$

where $\tau[k+i|k]$ represents the state at the time step $k+i$ predicted by the state at the time step k , M is the length of optimization horizon under the sampling period Δt , \mathbf{Q}_τ , \mathbf{Q}_u and $\mathbf{Q}_{\Delta u}$ denote the corresponding weight matrices. The first term of the objective function penalizes the divergence of the solved trajectory from the reference trajectory. The second and third term suppress the excessive control actions and control input rates (i.e., jerks and steering angle rates), respectively.

With the objective function and constraints introduced in the above subsections, the trajectory planning is formulated as an optimization problem, i.e.,

$$\begin{aligned} \min_{\mathbf{u}, \lambda, \mu, \rho} J_p \\ \text{s.t. } \tau[k|k] = \tau_{opt}^d[k], \end{aligned} \quad (25)$$

(17), (18), (23).

where the first constraint guarantees the initial solved state $\tau[k|k]$ to be equal to the reference state $\tau_{opt}^d[k]$.

E. Ipopt-based approach for trajectory planning

To solve the optimization problem (25), we propose an algorithm to plan the trajectory for a lane change based

on interior point optimizer (Ipopt) [50]. Ipopt is a widely used numerical solver for large-scale nonlinear optimization problems.

The algorithm for trajectory planning based on Ipopt is described in Algorithm 1. The optimization problem at the time step k is established as (25). By solving this problem, a sequence of optimal control actions $\mathbf{U}^*[k] = \{\mathbf{u}^*[k], \dots, \mathbf{u}^*[k+M-1]\}$ can be obtained. As the basic MPC algorithm in [51], we only apply the first control action $\mathbf{u}^*[k]$ to the ego vehicle to update its state. At the next time step $k+1$, the horizon moves forward and a new optimization problem is formulated and solved by our algorithm. By repeating this procedure, the ego vehicle can obtain an optimal trajectory that is smooth, collision-free, and satisfies the vehicle kinematic model. Till now we have formulated an optimization and an Ipopt-based algorithm to complete the task of trajectory planning.

Algorithm 1: Ipopt-based approach for trajectory planning

Data: Reference trajectory τ_{opt}^d , interactive trajectory prediction $\{\tau_{1,opt}^f, \dots, \tau_{n,opt}^f, \dots, \tau_{N,opt}^f\}$, ego vehicle's current state $\tau_{ego}^p[O]$
Result: Planned trajectory τ_{ego}^p

- 1 Initialization: $k = O$;
- 2 **while** $k \leq O + P - M$ **do**
- 3 Reset variables, and initial state constraint;
- 4 **for** $m = 1$ to M **do**
- 5 Reset predictive model (17);
- 6 Reset mechanical and physical constraints (18);
- 7 Calculate ego vehicle's \mathbf{A} and \mathbf{b} using (20);
- 8 **for** $n = 1$ to N **do**
- 9 Calculate \mathbf{A} and \mathbf{b} using $\tau_n^f[k+m]$ and (20);
- 10 Reset safety constraints (23);
- 11 **end**
- 12 **end**
- 13 Reset the objective function (24);
- 14 Solve above optimization using Ipopt;
- 15 Update ego vehicle's state using (17);
- 16 $k = k + 1$;
- 17 **end**

VI. SIMULATION STUDY

In this section, we show the effectiveness and flexibility of the proposed lane change method through simulations in different complex traffic scenarios and comparison with other methods. The effectiveness is further discussed by analyzing the prediction of other vehicles' interactive trajectories.

A. Simulation environment

An open-source automotive driving simulator [52] is used to implement the simulation scenarios. We consider two-lane or three-lane highway. All vehicles have the same shapes; i.e., 4 meters in length and 1.8 meters in width. Only the ego vehicle uses our proposed method for lane change. For other vehicles, the longitudinal motions and lateral lane change behaviors are governed by the Intelligent Driver Model (IDM) [53] and the Minimizing Overall Braking Induced by Lane changes (MOBIL) [54] model, respectively. In order to increase the

TABLE I
DEFAULT PARAMETER VALUES OF SIMULATION ENVIRONMENT.

Parameter	Description	Value
w	Vehicle width (m)	1.8
l	Vehicle length (m)	4.0
m	Vehicle mass (kg)	1412
L	Lane number	2 or 3
λ_s	Weight of safety	0.4
λ_e	Weight of efficiency	0.3
λ_c	Weight of comfort	0.3
$a_{y \max}$	Maximum lateral acceleration (m/s^2)	1.0
$\Delta a_{y \max}$	Maximum lateral jerk (m/s^3)	2.0
$a_{x \max}$	Maximum longitudinal acceleration (m/s^2)	4.0
$\Delta a_{x \max}$	Maximum longitudinal jerk (m/s^3)	1.0
\mathbf{u}_{\min}	Minimal control input	$[-4.0, -0.3]^T$
\mathbf{u}_{\max}	Maximum control input	$[4.0, 0.3]^T$
T_o	Observation horizon (s)	2.0
T_p	Prediction horizon (s)	1.0
T_c	Control horizon (s)	1.0
Δt	Sampling time (s)	0.1
$diag(\mathbf{Q}_\tau)$	Diagonal elements of \mathbf{Q}_τ	$[1, 10, 20, 1]^T$
$diag(\mathbf{Q}_u)$	Diagonal elements of \mathbf{Q}_u	$[0.1, 0.1]^T$
$diag(\mathbf{Q}_{\Delta u})$	Diagonal elements of $\mathbf{Q}_{\Delta u}$	$[0.1, 0.1]^T$
C_f	Cornering stiffness of front tire (N/rad)	55000
C_r	Cornering stiffness of rear tire (N/rad)	55000
I_z	Moment of inertia about z axis ($kg \cdot m^2$)	1536.7

complexity of the simulated scenarios, the parameters in the IDM and the MOBIL are randomly sampled from uniform distributions. Unless otherwise specified, our simulation uses the settings in Table I by default.

B. Performance of trajectory prediction

The accuracy of trajectory prediction has a significant influence on the decision making and trajectory planning, so we briefly discuss the performance of trajectory prediction in this subsection. Interested readers are referred to Ref. [38] for more details. To train GSTCN for our lane change method, we use the above simulator to generate 10,800 data segments under free, moderate, and heavy traffic scenarios. To be more realistic, we add random noise to simulate the measurement errors. Each segment contains 5-second trajectories, in which the observation horizon and the prediction horizon are set to be 2 seconds and 3 seconds, respectively. These segments are randomly split into training, validation, and testing sets.

To evaluate the performance of GSTCN, we compare it with two baselines.

- Constant Velocity Model (CV): This baseline uses the average velocity in the past time horizon to represent the future velocity of a vehicle.
- Long short-term memory network (LSTM): This baseline is a LSTM encoder-decoder that takes the past trajectory of the target vehicle as inputs and generates a future trajectory of the target vehicle.

As shown in Table II, GSTCN achieves the best prediction results than those of all baselines. Compared to CV, GSTCN and LSTM has an ability to learn nonlinear mapping from

TABLE II
COMPARISON OF RMSE FOR DIFFERENT TRAJECTORY PREDICTION MODEL. DATA ARE IN METERS.

Prediction Horizon(s)	CV	LSTM	Ours
1	0.43	0.38	0.27
2	0.98	0.81	0.72
3	1.62	1.35	0.89
Average	1.01	0.85	0.63

abundant data, so they outperform CV. In addition, since graph convolutional network is more effective to capture the spatial dependencies between vehicles than LSTM, GSTCN has a better accuracy than that of LSTM. Therefore, GSTCN can provide a reliable prediction for the subsequent decision making and trajectory planning.

C. Performance of proposed lane change method

1) *Baselines*: In order to further demonstrate the superiority of our proposed method, we compare it with the following baselines.

- **IDM+MOBIL**: This baseline controls the lateral and longitudinal motions with MOBIL and IDM, respectively, which is the same as other vehicles.
- **Our proposed method without interactive trajectory prediction (denoted as Proposed W/O IP)**: This baseline is the same as our proposed method except that it uses constant velocity, instead of the interactive trajectory prediction method in Section IV-B, to predict other vehicles' trajectories.

2) *Quantitative analysis*: In this subsection, we make quantitative analysis for the proposed method. We totally conduct 100 simulations under different traffic densities. In each simulation, vehicles run on a three-lane highway, and the parameters and initial positions for each vehicle are random. The mean velocities of the ego vehicle and surrounding vehicles are listed in Table III.

We can see that ego vehicles equipped with our proposed method achieve the fastest speeds, demonstrating the effectiveness of interactive trajectory prediction. Meanwhile the mean speed of other vehicles are faster than that of other baselines. This is because the ego vehicle also considers the driving efficiencies of surrounding vehicles when making decisions on lane changes. Therefore, their driving efficiencies will be improved when the ego vehicle cooperates with others. The ego vehicle using IDM+MOBIL only considers the observation at the current time but ignores the prediction of the future, so the mean speeds are the slowest.

3) *Case study*: As shown in Fig. 6, we also study two practical cases for lane change, including obstacle avoidance in dense traffic, and complex traffic with three lanes having different speeds and space gaps.

Case 1: In this case, as shown in Fig. 6(a), the ego vehicle drives in dense traffic, and the space gap between any two consecutive vehicles in Lane 1 is insufficient for others to

TABLE III
MEAN SPEEDS OF THE EGO VEHICLE AND OTHER VEHICLES UNDER 100 SIMULATIONS.

Object	IDM+MOBIL	Proposed W/O IP	Proposed
Ego vehicle (m/s)	13.95	14.85	16.01
Other vehicles (m/s)	14.19	14.31	14.56

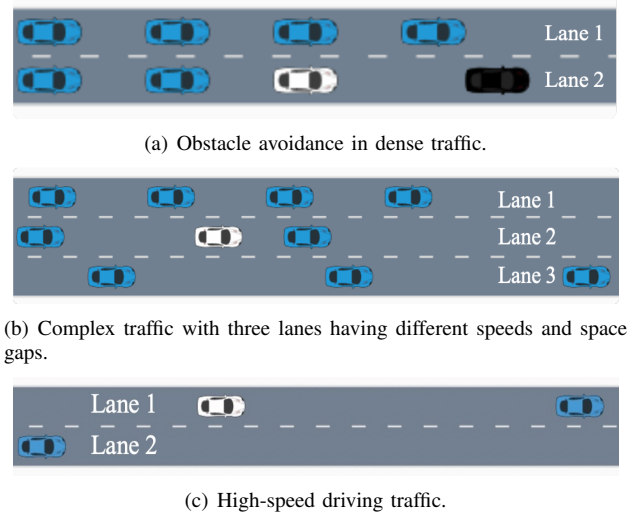


Fig. 6. Top views of the studied traffic scenarios. The white vehicle is the ego vehicle, the black one represents a stationary obstacle, and the blue vehicles are normal.

merge into. There is a stationary obstacle in front of the ego vehicle, so the ego vehicle should decide whether to stay in the current lane and stop or change to another lane for normal driving. If the ego vehicle chooses the latter decision for a higher efficiency, it needs the ability to interact with other vehicles to make them yield and make a space for it to cut in.

Fig. 7 shows the visualization of testing results at different time instants. It can be seen that although the initial inter-vehicle distances in Lane 1 are too narrow to merge in, the ego vehicle still successfully changes to that lane. As shown in Fig. 7(b), at the time instant $t = 2s$, the ego vehicle heads to the left and is going to overtake the left vehicle, which slows down to help the ego vehicle cut in. As shown in Fig. 7(c), at the time instant $t = 3s$, the ego vehicle has overtaken the left vehicle and is merging into the target lane. Finally, as shown in Fig. 7(e), the ego vehicle successfully changes to Lane 1 at about 8s.

Fig. 8 illustrates how the ego vehicle achieves interactions with surrounding vehicles by adjusting its acceleration and steering angle. The whole process for lane change can be divided into three stages: *before*, *during*, and *after merging*. Between 1~3s, it is *before merging*. The ego vehicle prepares for lane change by applying positive steering angle to approach the target lane and accelerating to overtake the left vehicle. When the ego vehicle is close enough to the target lane, as shown in Fig. 7(b), the gap in front of the left vehicle is larger than the initial one, demonstrating the left vehicle is slowing

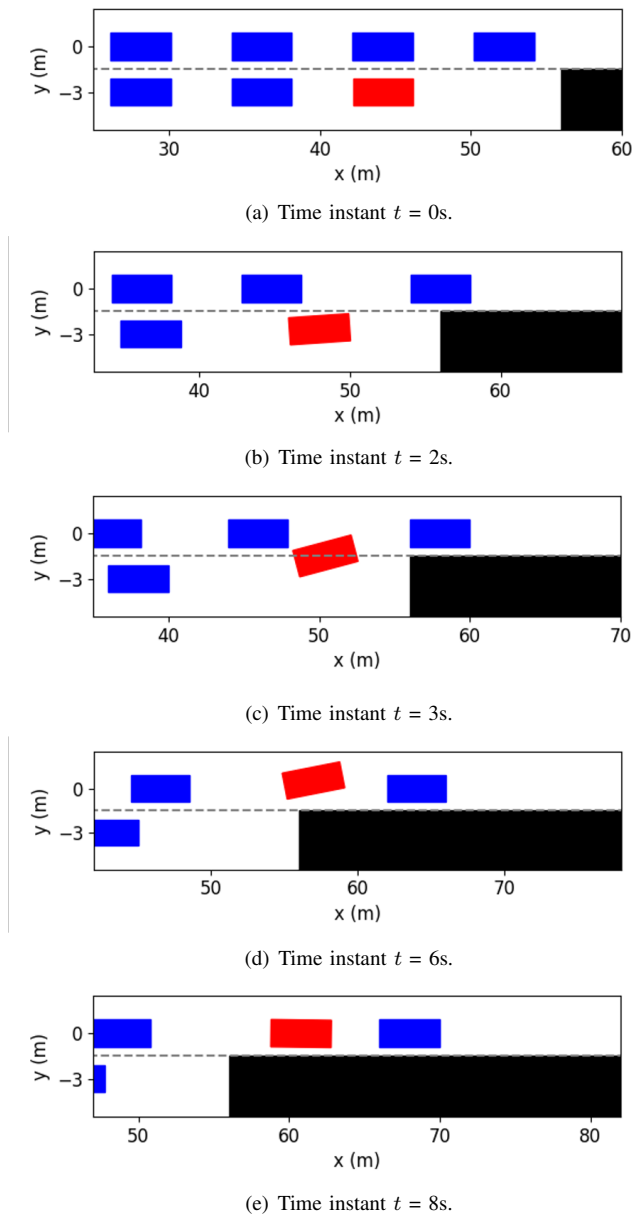


Fig. 7. Visualization of the testing results based on the proposed lane change method in the first case. Rectangles in red, blue, and black represent the ego vehicle, other vehicles, and the stationary obstacle, respectively. The gray dashed line indicates the lane boundary.

down and giving way to the ego vehicle. In the *during merging* stage during 3~7.8 s, the ego vehicle needs to perform intensive interactions with left vehicles to promote cooperation with them. The interactions are achieved by adjusting its velocity and heading angle to inform others, so we can see the acceleration and velocity curves have some fluctuations. In this process, the ego vehicle speeds up to overtake the left vehicle to merge into Lane 1, and must slow down to avoid collision whenever it is too close to the preceding vehicle. In the third stage, the ego vehicle has successfully entered the target lane. At this time, the ego vehicle only keeps the lane and does not need to interact with others to seek cooperation, so its acceleration and steering angle gradually tend to zeros.

To demonstrate the superiority of our proposed method, the

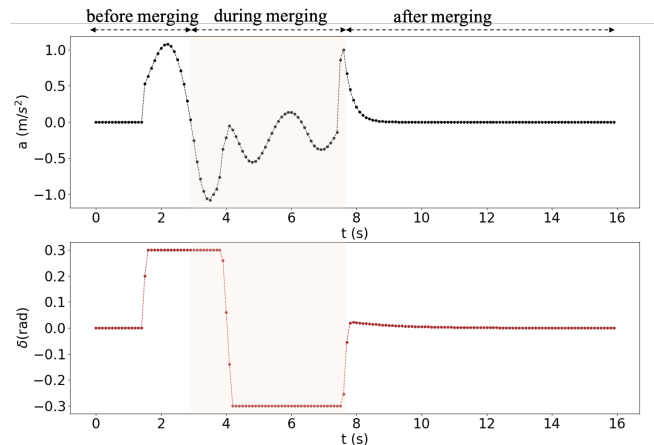


Fig. 8. The acceleration and steering angle of the ego vehicle in the first case.

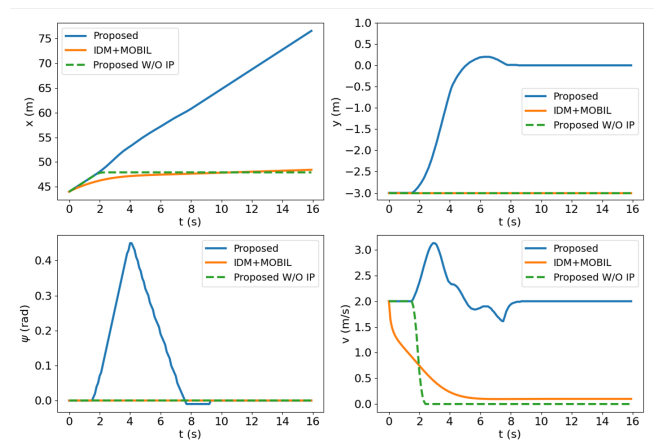


Fig. 9. Comparison of vehicle states generated by the proposed lane change method and baselines in the first case.

comparison between our method and the above baselines is shown in Fig. 9. We can see that both baselines choose a relatively conservative strategy. They decide to stay in the current lane, and thus the ego vehicle has to slow down to avoid collision with the obstacle. Since the other two algorithms cannot predict nearby vehicles' reactions to ego vehicle's decisions, they think lane change in that case is a dangerous behavior. To keep safe, the ego vehicle equipped with these two algorithms can only choose a conservative strategy, i.e., stopping and waiting. On the contrary, our proposed method accurately predicts nearby vehicles' reactions to ego vehicle's each decision, so it thinks lane change is an optimal decision in term of safety, efficiency, and comfort. During the trajectory planning, we also incorporated prediction as safety constraints, so the planned trajectory enables the ego vehicle to interact with surrounding vehicles to promote cooperation without compromising safety.

In order to further investigate how our method determines whether the lane change is a feasible decision, and how to use vehicular interactions to achieve cooperation during trajectory planning, Fig. 10 shows several examples of the interactive trajectory prediction in this case. Our proposed

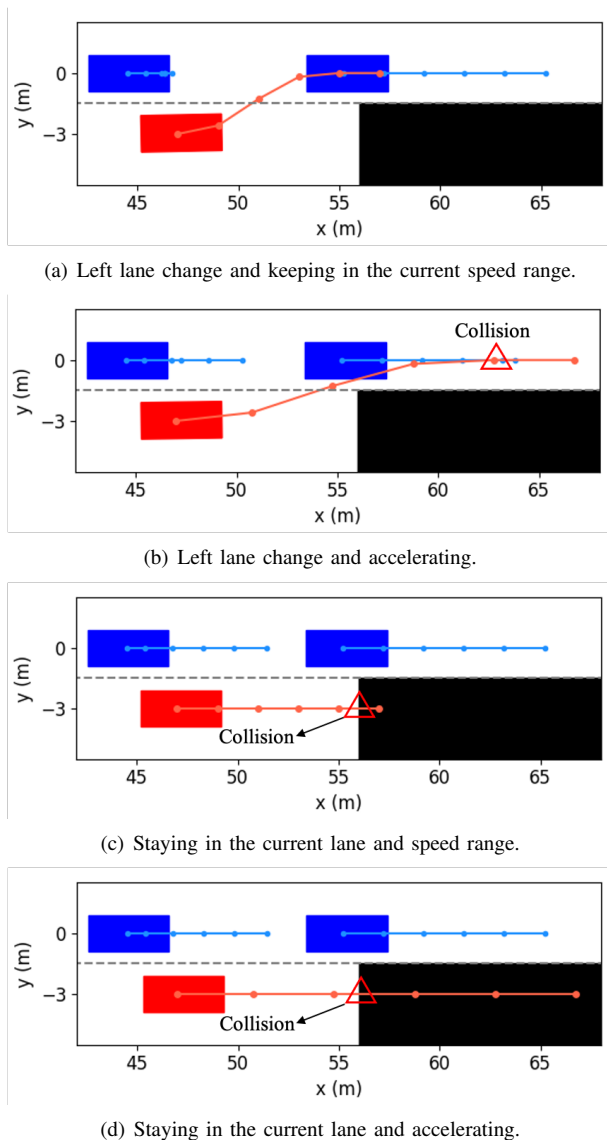


Fig. 10. Interactive trajectory prediction under different decisions in the first case. The blue lines represent the prediction, and the red lines are the reference trajectory generated by the method in Section IV-A.

method selects the decision in Fig. 10(a) as the optimal one, since the interactive trajectory prediction indicates that if the ego vehicle runs in this way, the vehicle behind will slow down and enlarge the gap. Therefore, this decision can help the ego vehicle avoid collision with the obstacle and change lanes safely to ensure normal driving. The decision in Fig. 10(b) can also make the ego vehicle overtake the left vehicle and then change lanes, but the prediction shows that the ego vehicle would collide with the front vehicle. This is consistent with the intuition that the front vehicle rarely reacts to the speed of the following vehicle. We can see from Fig. 10(c) and (d) that if the ego vehicle neither changes to another lane nor decelerates, it will collide with obstacles. Based on the above discussion and analysis, we conclude that our proposed method is able to obtain reliable interactive trajectory prediction, which can be further used to determine the most proper decision and provide the motion planning module with safety constraints to avoid

collision between vehicles.

Note that with communication technologies (e.g., C-V2X, DSRC), the rear vehicle could also give its way. However, in this paper, we only use the information perceived by the ego vehicle. There are two reasons for this choice. First, vehicular communication is not applicable to all scenarios as some vehicles may not install communication equipment, or the equipment fails in some cases. In this regard, the methods relying on communication cannot make the rear vehicle yield. Since the raw data we get only relies on the sensors of the ego vehicle, our method can be used in more scenarios than communication-based methods. In addition, our method does not require all vehicles to be equipped with communication facilities, so it is easier to implement. Second, even if all vehicles are equipped with communication equipment, there is no guarantee that these vehicles will cooperate with each other. This is because automated vehicles manufacturers from different companies usually only consider their own interests, the algorithms they use may cause competition instead of cooperation.

Case 2: In this case, a complex traffic scenario is studied. Fig. 6(b) shows the top view snapshot of vehicles at the initial time. The vehicles in Lane 1 have the fastest speeds and narrowest inter-vehicle distances, so it is risky to change to this lane in order to obtain a higher speed. On the contrary, vehicles in Lane 3 have the slowest speeds and largest inter-vehicle distances. The ego vehicle is driving in Lane 2, and vehicles in Lane 2 have moderate speeds. Since the vehicle in front of the ego vehicle travels at a slower speed than vehicles in Lane 1, the ego vehicle needs to decide whether to follow the preceding vehicle and keep a slow speed to guarantee the safety or to change to Lane 1 to gain a higher speed. The decision is dependent on the reaction of surrounding vehicles and whether a feasible trajectory can be generated.

The simulation results at different time instants are presented in Fig. 11. We can see that the ego vehicle makes two lane changes to reach the fastest speed. At first, in order to get rid of the influence of the preceding vehicle with slow speeds, the ego vehicle tries to change to Lane 1 by interacting with other vehicles. As shown in Fig. 11(b), at the time instant $t = 2s$, the left following vehicle slows down to expand the space gap to help the ego vehicle to cut in. As shown in Fig. 11(c), the ego vehicle completes lane change and overtaking at the time instant $t = 4s$. Further, as shown in Fig. 11(f), the ego vehicle returns to Lane 2 and obtains the fastest speed.

These interactive behaviors can also be interpreted by the acceleration and steering angle of the ego vehicle in Fig. 12. Similar to the first case, the process for two lane changes is divided into six stages. Since the initial speed of the ego vehicle is lower than the left vehicle's, it increases its longitudinal speed while moving towards Lane 1 *before the first merging*. At about 2 to 4 seconds, the acceleration and steering angle of the ego vehicle fluctuate, which indicates that the ego vehicle wants to switch to the target lane quickly while avoiding collision with the preceding vehicle. Between 4~6.7s, the control actions are getting closer to zeros since the ego vehicle has finished the first lane change and only needs to drive steadily. At about 7 seconds, as shown in

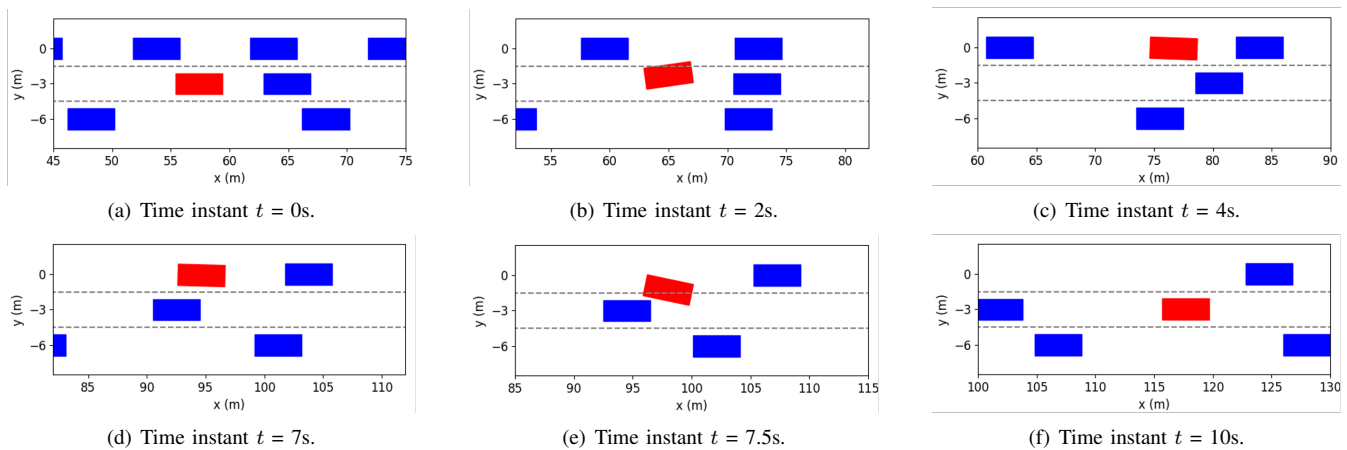


Fig. 11. Visualization of the testing results based on the proposed lane change method in the second case.

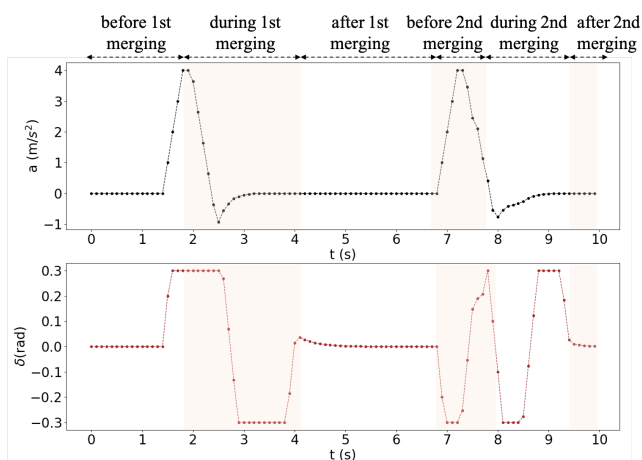


Fig. 12. The acceleration and steering angle of the ego vehicle in the second case.

Fig. 11(d), the space gap between the front vehicle and the right one is large enough, and there is no front vehicle in the Lane 2. Consequently, after the prediction and evaluation, our proposed method decides to accelerate and change to Lane 2. Between 8~9.5s, the ego vehicle smoothly enters the target lane.

As shown in Fig. 13, in the second studied case, the ego vehicle using both baselines exhibits conservative behaviors. For the baseline Proposed W/O IP, we can see that it decides to stay in the current lane and speed. Because of the lack of interactive prediction, this baseline believes there would be a collision when changing to Lane 1. For another baseline, we can see that at about 0.5 seconds, the ego vehicle attempts to switch to Lane 3 where the space gap between vehicles is large. But soon it finds that the speeds of vehicles in this lane are too slow, so it returns to the original lane. This is because this baseline cannot predict the future trajectory of other vehicles. Therefore, when it decides to change to Lane 3, it only considers the benefits of a larger space gap, but cannot predict the future speed loss. The lack of trajectory prediction leads to large amplitudes and frequent actions on the steering wheel, which reduces the driving comfort. After returning to

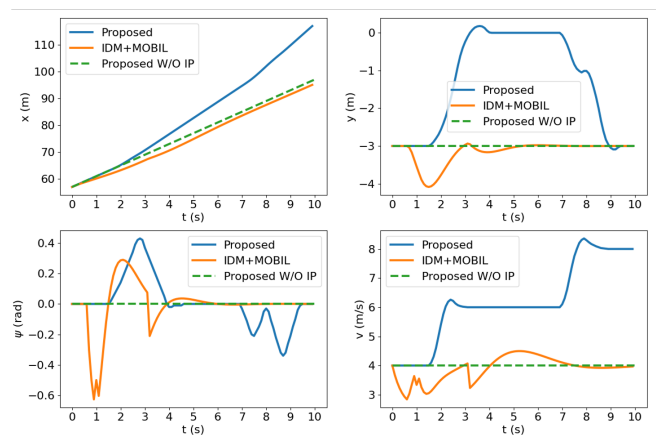


Fig. 13. Comparison of vehicle states generated by the proposed lane change method and baselines in the second case.

the original lane, the ego vehicle has to follow the preceding vehicle and maintain a relatively slow speed. Similar to the first case, although the speeds of vehicles in Lane 1 are faster, the ego vehicle is not recommended to change that lane because the close distances between vehicles may cause a collision. However, the ego vehicle equipped with our proposed method is more inclined to create a favorable environment through active cooperation with other vehicles, so it can achieve safe, comfortable and efficient driving.

We take the first merging as an example to explore how the proposed method decides to accelerate and change to Lane 1. The interactive trajectory prediction for each decision is shown in Fig. 14, and the cost values are listed in Table IV. From the prediction in Fig. 14(f), (h) and (i), we can see that these three decisions would lead to collisions, and thus their cost values for safety are much higher than others'. When the ego vehicle changes to the left lane at low or current speeds, the prediction in Fig. 14(a) and (b) indicate that the left following vehicle will slow down. Despite both decisions are safe, their cost values for efficiency vary greatly, resulting in a large difference in the total costs. Besides, although keeping in the current lane and speed range would not cause any costs for safety and comfort, its efficiency cost contributes to a higher total cost

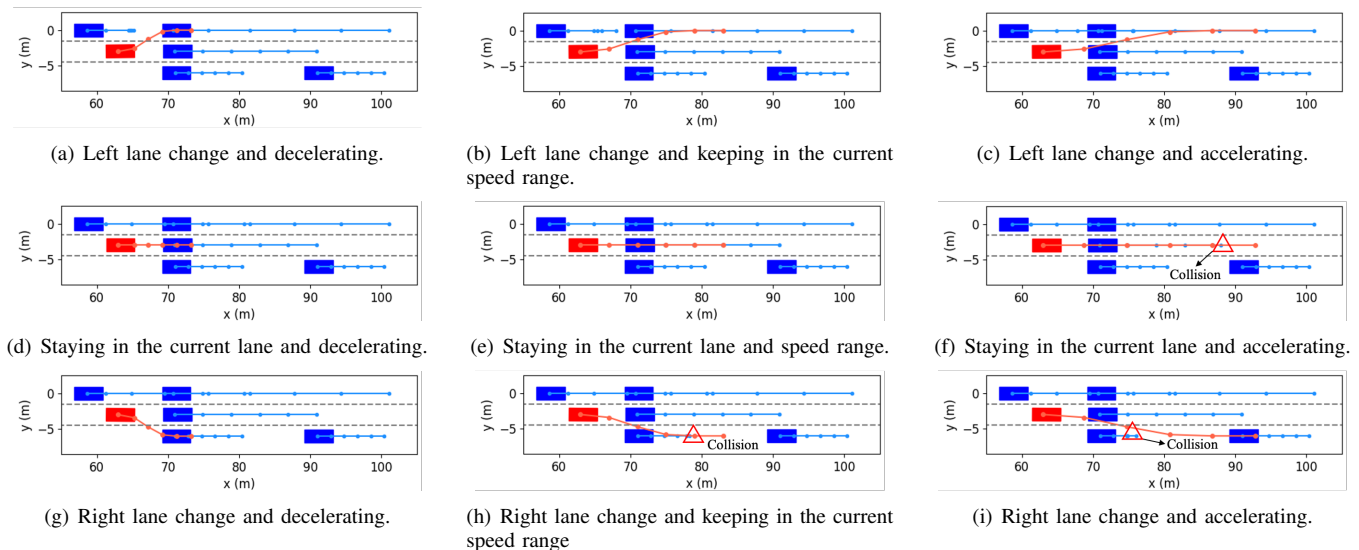


Fig. 14. Interactive trajectory prediction under different candidate decisions in the first decision process of the second case.

TABLE IV
COST VALUES FOR DIFFERENT CANDIDATE DECISIONS IN THE FIRST
DECISION PROCESS OF THE SECOND CASE.

Decision	Safety	Comfort	Efficiency	Total cost
Left lane change & decelerating	39.6	388.3	8094.8	8522.7
Left lane change & speed keeping	23.6	262.0	5020.4	5306.0
Left lane change & accelerating	749.8	388.3	1633.2	2771.3
Lane keeping & decelerating	290.0	126.3	6293.1	6709.4
Lane keeping & speed keeping	0.0	0.0	3600.0	3600.0
Lane keeping & accelerating	8139.0	126.3	1669.1	9934.4
Right lane change & decelerating	211.8	388.3	6094.8	6694.9
Right lane change & speed keeping	7468.0	262.0	3520.4	11250.4
Right lane change & accelerating	3051.3	388.3	1633.2	5072.8

than that of left lane change & accelerating. Therefore, our proposed method selects left lane change & accelerating as the optimal decision. From the results in Fig. 11~13, we can see that the planned trajectory is safe and smooth, has low requirements for control actions, and achieves higher driving efficiency, demonstrating it is an appropriate decision.

Case 3: The first two cases verify the effectiveness of our proposed method in extreme scenarios, so we conduct this case study to further evaluate the performance in routine highway scenarios (e.g., high-speed driving). As shown in Fig. 6(c), three vehicles are driving on a two-lane highway. The ego vehicle is driving on Lane 1, and there are two vehicles in the front and right side of the ego vehicle. Since the front vehicle drives with a slow velocity, the ego vehicle needs to decide

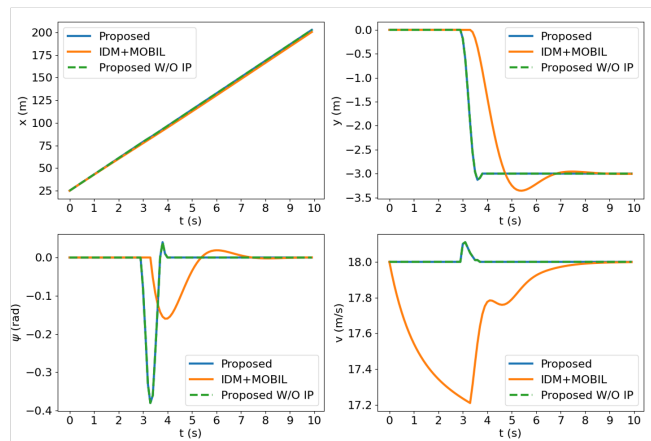


Fig. 15. Comparison of vehicle states generated by the proposed lane change method and baselines in the third case.

whether to slow down to keep a safe distance or to change lanes to overtake. The initial vehicle states are set as follows. The longitudinal velocities of ego, front, and right vehicles are set to 18 m/s, 16 m/s, and 15 m/s. Their initial locations are (25.0, 0.0), (55.0, 0.0), and (10.0, -3.0), respectively.

Fig. 15 shows the comparison results between our method and baselines. We can see that three methods have similar performance. They all changed to Lane 2 at similar times. In particular, Proposed W/O IP has the same planned trajectory as the proposed method. This is because the space gaps between vehicles in routine high-speed scenarios are larger and the lane changes usually involve fewer or no interactions than the first two cases. Therefore, our proposed method can handle both routine and extreme lane changes.

VII. DISCUSSION

Finally, we analyze the effects of different hyperparameter values, and the necessity of considering module interdependence.

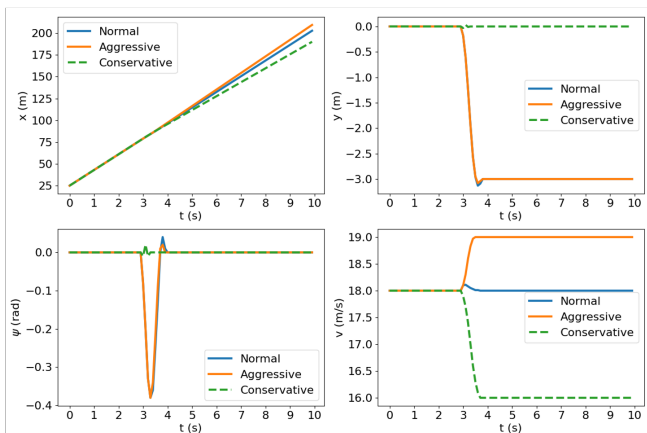


Fig. 16. Comparison of vehicle states generated by the proposed lane change method with different driving styles.

A. Hyperparameter analysis

In this subsection, we discuss the effects of different hyperparameter values on our proposed method.

1) *Decision weight*: When making decisions, the ego vehicles are supposed to satisfy the preference of different drivers and passengers under different traffic scenarios. For example, safer and more comfortable driving is desirable when the elderly or children are in the vehicle. In contrast, if drivers and passengers are in a hurry, they might prioritize travel efficiency. Therefore, to make the decisions personalized for different driving scenarios, we discuss the effects of different decision weights on the planning. We define three driving styles, i.e., conservative, aggressive, and normal [55]. The decision weight values in Table I represent the normal style. The aggressive drivers assign weights of 0.3, 0.4, 0.3 to safety, efficiency, and comfort. Conservative drivers assign the corresponding weights to 0.6, 0.2, 0.2.

We use Case 3 to illustrate the effects of different decision weights. As shown in Fig. 16, we can see that different driving styles would lead to various decisions and motions. The ego vehicle with aggressive or normal style changes to Lane 2 with a high speed, while the conservative style would decelerate and follow the front car. This is because the cost of lane keeping for the ego with conservative style is lower than lane change. However, the ego with an aggressive or normal style gives a higher priority to travel efficiency, so it chooses to change lanes to keep a high speed. Besides, the aggressive vehicle has a faster speed after lane change than the normal one, which is consistent with the higher weight on efficiency.

2) *Prediction and control horizons*: Prediction horizon and control horizon are two important parameters of MPC. The prediction horizon represents how many time steps to predict the future states, and the control horizon defines how many groups of control actions are needed to be solved. Note that the prediction horizon should be greater than or equal to the control horizon. As shown in Table V, we use Case 3 to discuss the effects of prediction and control horizons on the objective value and computation time.

The objective value reflects the deviation between the planned trajectory and the reference, and the amplitudes of

TABLE V
OBJECTIVE VALUE / COMPUTATION TIME USING DIFFERENT PREDICTION HORIZONS AND CONTROL HORIZONS. THE FIRST COLUMN CORRESPONDS TO THE PREDICTION HORIZON. THE FIRST ROW CORRESPONDS TO THE CONTROL HORIZON.

	0.5	1.0	1.5
0.5	17.1 / 26	-	-
1.0	18.4 / 40	17.5 / 43	-
1.5	20.7 / 43	17.7 / 45	17.7 / 62

the control actions and control input rates. We can see that the objective value increases with the prediction horizon but decreases with the control horizon. Computational efficiency is an important factor when considering the applicability of a method. All the above simulations were implemented in the Julia programming language and performed on a 2017 MacBook Pro with an Intel i5 processor clocked at 2.3 GHz. We can see that the computation time increases with both the prediction horizon and the control horizon, and all computation times are less than the sampling time $\Delta t=100$ ms. It is worth noting that although the computation time increases with the complexity of traffic scenarios, it can be greatly shortened by using C++ to optimize the codes when deployed in real cars.

B. Module interdependence analysis

Previous methods usually take the decision making and trajectory planning for lane change as two separate problems. However, as a high level module, decision maker usually outputs more general instructions, which may cause conflicts (e.g., excessive acceleration) with the low level trajectory planner. Thus, we discuss the necessity of considering module interdependence. To this end, we first define a baseline, i.e., our proposed method without considering interdependence (Proposed W/O ID). This baseline is the same as our proposed method except that its decision making module does not consider the constraints of trajectory planning. Then, we slightly modify the setting of Case 3 to illustrate this problem. The initial location of the right vehicle is changed to (25.0, -3.0), and its speed is changed to 18m/s. In this way, the distance and speed difference between the right vehicle and the ego vehicle become smaller. Finally, the maximum and minimum accelerations of the baseline's trajectory planning module are set to be half of the original values.

As shown in Fig. 17, we can see the Proposed W/O ID takes more time to change lanes with a less smooth trajectory. This is because the accelerations used in decision making and planning are inconsistent. The decision maker believes that a lane change can be successfully achieved using a large acceleration. However, because only half maximum acceleration can be used by the trajectory planner, the ego vehicle fails to accelerate to the desired speed in time. Finally, it takes more time to change lanes, and the trajectory is less smooth. On the contrary, our proposed method considers the consistency between decision making and trajectory planning, so the decision ensures the feasibility of trajectory planning.

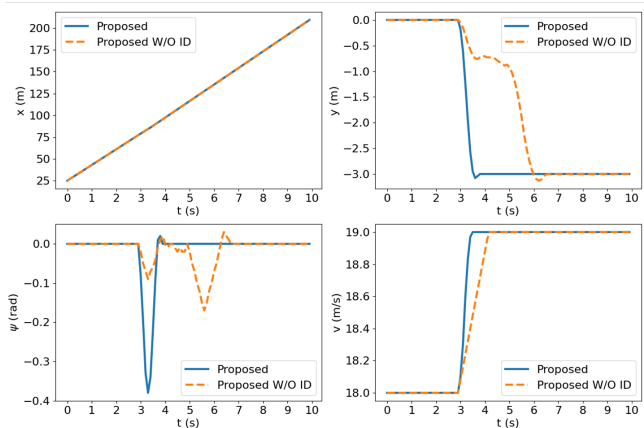


Fig. 17. Comparison of vehicle states generated by the proposed lane change method and Proposed W/O ID.

C. Path following

In this subsection, we demonstrate that the planned trajectory can be used by a more complex model of the control module. To be more realistic, we use a vehicle dynamic model in Ref. [56] to consider the lateral tire forces of the front and rear wheels, respectively. The state space model is defined as,

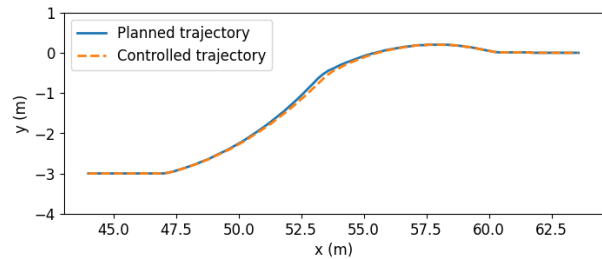
$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = A \cdot \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_f}{m} \\ 0 \\ \frac{2l_f C_f}{I_z} \end{bmatrix} \delta, \quad (26)$$

where,

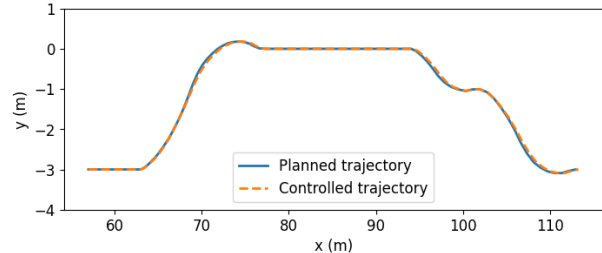
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_f + 2C_r}{mv_x} & 0 & -v_x - \frac{2C_f l_f - 2C_r l_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2l_f C_f - 2l_r C_r}{I_z v_x} & 0 & -\frac{2l_f^2 C_f + 2l_r^2 C_r}{I_z v_x} \end{bmatrix}, \quad (27)$$

C_f and C_r are the cornering stiffness of each front tire and rear tire, m is the mass of a vehicle, I_z denotes the moment of inertia about the z axis. The values of these parameters are listed in Table I. More details can be found in Ref. [56]. We use MPC to solve the trajectory controlled by (26) using the planned trajectories of the cases in Section VI-C3. Case 1 is in a dense traffic where the ego vehicle needs to perform obstacle avoidance. Case 2 consists of a complex traffic with three lanes having different speeds and space gaps, where the ego vehicle needs to change lanes to improve the driving efficiency. Case 3 is in a routine highway scenario, where the ego vehicle changes lanes to maintain a high speed.

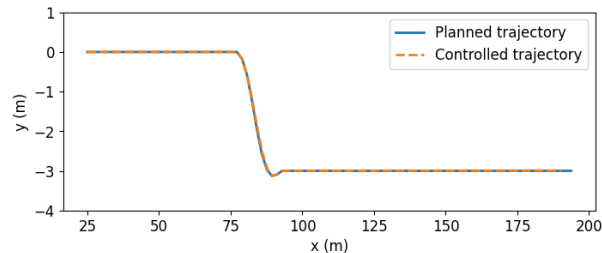
As shown in Fig. 18, we use the planned trajectories of the above three cases to evaluate our proposed method. We find that the controlled trajectories fit the planned trajectories well, demonstrating that our proposed method is able to control a more complex dynamic model. Even if the planned trajectories of the cases have different steering and speeds, the control module can follow the planned trajectories to control the vehicle dynamic model.



(a) Case 1 in Section VI-C3: obstacle avoidance in dense traffic.



(b) Case 2 in Section VI-C3: complex traffic with three lanes having different speeds and space gaps.



(c) Case 3 in Section VI-C3: high-speed driving traffic.

Fig. 18. Path following of planned trajectories using vehicle dynamic model.

VIII. CONCLUSIONS

In this paper, we have presented a cooperation-aware method that can help the ego vehicle to execute a proper lane change for guarantee driving efficiency and safety. Different from the existing methods, we have considered the interdependence between the lane change decision module and the trajectory planning module so as to avoid possible conflicts between them. An interactive trajectory prediction method has been proposed, with which we have designed a candidate evaluation method. By doing so, our approach can find potential cooperation by predicting others' interactions so that a less conservative decision on lane change can be made. With a right lane change decision, a safe, efficient, and comfortable driving can be well guaranteed. During the process of planning a lane change trajectory, interactive trajectories has been incorporated into constraints, such that the ego vehicle can interact with others to promote cooperation by adjusting its control inputs. The quantitative results have demonstrated that when using our proposed method, the driving efficiencies of the ego vehicle and others are 14.8% and 2.6% higher than those with methods without interactive prediction. Two practical case studies further analyze how the proposed method predicts interactive trajectories of other vehicles and uses this interactive prediction to make decisions.

Following the stream of this study, several future works can be done. (i) The uncertainty caused by communication delays can be considered to improve the robustness. (ii) The proposed lane change method is only applied on single vehicle to improve driving efficiencies of its nearby vehicles and itself. Based on our method, exploring a distributed framework to optimize a large traffic network is also worthwhile for future research.

REFERENCES

- [1] 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 Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.
- [2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [3] J. Wei, J. M. Dolan, and B. Litkouhi, "A prediction-and cost function-based algorithm for robust autonomous freeway driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 512–517.
- [4] S. Moridpour, M. Sarvi, G. Rose, and E. Mazloui, "Lane-changing decision model for heavy vehicle drivers," *J. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 24–35, 2012.
- [5] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2063–2067.
- [6] V. Sezer, "Intelligent decision making for overtaking maneuver using mixed observable markov decision process," *J. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 201–217, 2018.
- [7] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018.
- [8] Y. Hou, P. Edara, and C. Sun, "Modeling mandatory lane changing using Bayes classifier and decision trees," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 647–655, Apr. 2014.
- [9] Y. Guan, D. Li, S. Xue, and Y. Xi, "Feature-fusion-kernel-based Gaussian process model for probabilistic long-term load forecasting," *Neurocomputing*, vol. 426, pp. 174–184, Feb. 2021.
- [10] Q. Guan, Z. Sheng, and S. Xue, "HRPose: Real-time high-resolution 6d pose estimation network using knowledge distillation," *Chin. J. Electron.*, vol. 32, no. 1, pp. 189–198, Jan. 2023.
- [11] X. Gu, Y. Han, and J. Yu, "A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost," *IEEE Access*, vol. 8, pp. 9846–9863, Jan. 2020.
- [12] J. Zhang, Y. Liao, S. Wang, and J. Han, "Study on driving decision-making mechanism of autonomous vehicle based on an optimized support vector machine regression," *Appl. Sci.*, vol. 8, no. 1, p. 13, Dec. 2018.
- [13] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE Access*, vol. 7, pp. 26 543–26 550, Feb. 2019.
- [14] J. Zheng, K. Suzuki, and M. Fujita, "Predicting driver's lane-changing decisions using a neural network model," *Simul. Model. Pract. Theory*, vol. 42, pp. 73–83, Mar. 2014.
- [15] L. Li, K. Ota, and M. Dong, "Humanlike driving: Empirical decision-making system for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6814–6823, Aug. 2018.
- [16] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 509–522, Jun. 2011.
- [17] X. Xu, L. Zuo, X. Li, L. Qian, J. Ren, and Z. Sun, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 10, pp. 3884–3897, 2020.
- [18] C. You, J. Lu, D. Filev, and P. Tsiotras, "Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1227–1232.
- [19] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [20] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 1994, pp. 3310–3317.
- [21] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field D* algorithm," *J. Field Robot.*, vol. 23, no. 2, pp. 79–101, 2006.
- [22] L. Parungao, F. Hein, and W. Lim, "Dijkstra algorithm based intelligent path planning with topological map and wireless communication," *ARPN J. Eng. Appl. Sci.*, vol. 13, no. 8, pp. 2753–2763, 2018.
- [23] D. Fassbender, A. Mueller, and H.-J. Wuensche, "Trajectory planning for car-like robots in unknown, unstructured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 3630–3635.
- [24] Y. Guan, K. Yokoi, O. Stasse, and A. Kheddar, "On robotic trajectory planning using polynomial interpolations," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Jul. 2005, pp. 111–116.
- [25] X. He, D. Xu, H. Zhao, M. Moze, F. Aioun, and F. Guillemard, "A human-like trajectory planning method by learning from naturalistic driving data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 339–346.
- [26] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 1036–1042.
- [27] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using bézier curves and MPC," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 826–833.
- [28] D. Yang, D. Li, and H. Sun, "2D Dubins path in environments with obstacle," *Math. Problems Eng.*, vol. 2013, pp. 1–6, Nov. 2013.
- [29] E. Bertolazzi, P. Bevilacqua, F. Biral, D. Fontanelli, M. Frego, and L. Palopoli, "Efficient re-planning for robotic cars," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1068–1073.
- [30] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, Apr. 2000, pp. 995–1001.
- [31] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2007, pp. 1617–1624.
- [32] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *J. Intell. Robot. Syst.*, vol. 71, no. 2, pp. 231–253, Aug. 2013.
- [33] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.
- [34] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 2061–2067.
- [35] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, "An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14 458–14 469, Dec. 2020.
- [36] Y. Lu, X. Xu, X. Zhang, L. Qian, and X. Zhou, "Hierarchical reinforcement learning for autonomous decision making and motion planning of intelligent vehicles," *IEEE Access*, vol. 8, pp. 209 776–209 789, Oct. 2020.
- [37] L. Guo, P. S. Ge, M. Yue, and Y. B. Zhao, "Lane changing trajectory planning and tracking controller design for intelligent vehicle running on curved road," *Math. Probl. Eng.*, vol. 2014, pp. 1–9, Jan. 2014.
- [38] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17 654–17 665, Oct. 2022.
- [39] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 1–12, Apr. 2021.
- [40] J. C. Hayward, "Near miss determination through use of a scale of danger," 1972.
- [41] L. M. Hynes and J. P. Dickey, "The rate of change of acceleration: Implications to head kinematics during rear-end impacts," *Accident Anal. Prevention*, vol. 40, no. 3, pp. 1063–1068, May 2008.
- [42] P. R. Grant and B. Haycock, "Effect of jerk and acceleration on the perception of motion strength," *J. Aircraft*, vol. 45, no. 4, pp. 1190–1197, Jul. 2008.
- [43] A. Sadat, M. Ren, A. Pokrovsky, Y.-C. Lin, E. Yumer, and R. Urtasun, "Jointly learnable behavior and trajectory planning for self-driving vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 3949–3956.

- [44] S. Bae, D. Saxena, A. Nakhaei, C. Choi, K. Fujimura, and S. Moura, "Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 1209–1216.
- [45] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1094–1099.
- [46] Z. Sheng, S. Xue, Y. Xu, and D. Li, "Real-time queue length estimation with trajectory reconstruction using surveillance data," in *Proc. 16th Int. Conf. Control, Autom., Robot., Vis. (ICARCV)*, Dec. 2020, pp. 124–129.
- [47] C. Ju, Z. Wang, C. Long, X. Zhang, and D. E. Chang, "Interaction-aware Kalman neural networks for trajectory prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1793–1800.
- [48] R. Firoozi, X. Zhang, and F. Borrelli, "Formation and reconfiguration of tight multi-lane platoons," *Control Eng. Pract.*, vol. 108, p. 104714, Mar. 2021.
- [49] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 972–983, May 2021.
- [50] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.
- [51] E. F. Camacho and C. B. Alba, *Model Predictive Control*. Springer, 2013.
- [52] Stanford Intelligent Systems Laboratory, Automotivedrivingmodels.jl, <https://github.com/sisl/AutomotiveDrivingModels.jl>.
- [53] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.
- [54] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec.*, vol. 1999, no. 1, pp. 86–94, 2007.
- [55] L. Xu, J. Hu, H. Jiang, and W. Meng, "Establishing style-oriented driver models by imitating human driving behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2522–2530, Oct. 2015.
- [56] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.