

Article

A Personalized Motion Planning Method with Driver Characteristics in Longitudinal and Lateral Directions

Di Zeng¹, Ling Zheng^{1,*}, Yinong Li¹, Jie Zeng² and Kan Wang²

¹ State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China; 20156244@cqu.edu.cn (D.Z.); ynli@cqu.edu.cn (Y.L.)

² China Merchants Testing Vehicle Technology Research Institute Co., Ltd., Chongqing 401329, China; cjzengjie@cmhk.com (J.Z.); wangkan@cmhk.com (K.W.)

* Correspondence: zling@cqu.edu.cn

Abstract: Humanlike driving is significant in improving the safety and comfort of automated vehicles. This paper proposes a personalized motion planning method with driver characteristics in longitudinal and lateral directions for highway automated driving. The motion planning is decoupled into path optimization and speed optimization under the framework of the Baidu Apollo EM motion planner. For modeling driver behavior in the longitudinal direction, a car-following model is developed and integrated into the speed optimizer based on a weight ratio hypothesis model of the objective functional, whose parameters are obtained by Bayesian optimization and leave-one-out cross validation using the driving data. For modeling driver behavior in the lateral direction, a Bayesian network (BN), which maps the physical states of the ego vehicle and surrounding vehicles and the lateral intentions of the surrounding vehicles to the driver's lateral intentions, is built in an efficient and lightweight way using driving data. Further, a personalized reference trajectory decider is developed based on the BN, considering traffic regulations, the driver's preference, and the costs of the trajectories. According to the actual traffic scenarios in the driving data, a simulation is constructed, and the results validate the human likeness of the proposed motion planning method.

Keywords: automated vehicles; motion planning; driver behavior



Citation: Zeng, D.; Zheng, L.; Li, Y.; Zeng, J.; Wang, K. A Personalized Motion Planning Method with Driver Characteristics in Longitudinal and Lateral Directions. *Electronics* **2023**, *12*, 5021. <https://doi.org/10.3390/electronics12245021>

Academic Editor: Mahmut Reyhanoglu

Received: 10 November 2023

Revised: 11 December 2023

Accepted: 12 December 2023

Published: 15 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Humanlike driving, which refers to making driving decisions according to the driver's preferences, is significant for automated vehicles (AVs). This is because driving behaviors without considering human driver characteristics are likely to not only cause physiological and psychological discomfort to the driver but also affect the cognition, operation, and motion state of other traffic participants, thus endangering traffic safety [1]. To give AVs the characteristics of human drivers, researchers developed automated driving systems based on various methods using naturalistic human driving data. However, few of them achieve highly humanlike driving that is predictable, interpretable, situation-aware, and interaction-aware [2,3].

The core task for achieving humanlike driving is driver behavior modeling, which can be divided into three groups: intention estimation, trait estimation, and motion prediction [4]. Intention estimation mainly refers to inferring what a driver might intend to do in the immediate future at a tactical level (e.g., change lane, follow) or an operational level (e.g., acceleration, deceleration) [5]. Trait estimation involves extracting features or patterns from human driving data, contributing to understanding the traits of driving behaviors. Motion prediction is the task of predicting the future physical states of other vehicles. Among the three groups, intention estimation, which can be further classified into longitudinal and lateral intention estimation according to the vehicle control direction, is directly related to building a humanlike automated driving system [2]. Note that the intention estimation in this paper refers to modeling the intention inference mechanism of

drivers instead of predicting drivers' intentions according to electroencephalograph signal, steering wheel angle, etc.

Regarding longitudinal intention estimation, most existing models were developed based on the dynamics method, for example, the intelligent driver model (IDM) [6], Wiedemann car-following model [7], linear car-following (LCF) model [8], modified linear car-following (MLCF) model [9], and three-state space car-following model [10]. These methods model driver behavior in the longitudinal direction by designing control models to control the acceleration and deceleration according to the motion states of the preceding vehicle. They perform car following well with clear mathematical structures. Recently, some researchers proposed establishing accurate and generalizable driver behavior models through inverse reinforcement learning (IRL) to achieve personalized adaptive cruise control (PACC) [11,12]. Compared with the traditional model-driven methods, IRL-based methods may obtain more humanlike models but reduce model interpretability.

For lateral intention estimation, model-driven methods, such as fuzzy theory [13–15] and model predictive control (MPC) [16], were utilized to mimic human reasoning. However, these methods are typically poorly humanlike and have difficulty handling the diversity of traffic scenes. To tackle this problem, data-driven methods were employed. In [17], the authors adopted convolutional neural networks (CNNs) and deep learning (DL) to directly map raw pixels to steering commands. Chen developed a deep planning model that combines a CNN with the long short-term memory (LSTM) module to make humanlike decisions [18]. However, directly learning the mapping between states and intentions suffers from the drawback of poor generalization [19]. Different from DL, IRL aims to obtain the parameters of the reward function that best explains human behavior using human demonstrations. Rosbach combined IRL and sample-based policy searching to obtain a reward function, based on which a final humanlike driving policy was selected from the discrete policy set [20]. One limitation of this method can be the heavy computational burden of policy searching. In [21], the authors also used IRL to model the lateral behavior of the driver. The difference is that candidate trajectories were generated based on polynomial curves, and the agent of IRL only decided the best trajectory, which preserved its predictability.

In summary, most existing intention estimation methods separately modeled driver behavior in either the longitudinal direction (e.g., [9,12]) or the lateral direction (e.g., [13,15,16]). Meanwhile, for high-level AVs, the driving task involves both car following and lane changing, which requires estimating both the driver's longitudinal and lateral intentions in the same humanlike automated driving system. Although some recent data-driven methods, e.g., [17,18,21], can model both car-following and lane-changing behaviors, they ignore the driver characteristics at the speed and acceleration levels in the longitudinal direction. One solution is to build a personalized lane-changing intention inference model and combine it with a car-following model. However, the limitations are (1) designing two control models increases development costs, and (2) vehicle dynamics constraints may not be satisfied for a vehicle controlled by two independent controllers. Furthermore, most driver models, such as [15,17,20,22], did not consider the interaction behavior of obstacle vehicles (e.g., lane-changing behavior of obstacle vehicles) in the inputs of the lateral intention inference models. This could be detrimental to AVs sharing the road with other vehicles. For instance, when a vehicle is about to change to a target lane, it may have to abort lane changing if the AV ignores its intention and changes to the same target lane. Moreover, the opaqueness of machine learning models prevents a user or developer from being able to verify, interpret, and understand the reasoning of the system. Additionally, (inverse) reinforcement learning has the disadvantages of low training efficiency and the feasibility question of transferring the knowledge acquired from ideal observations to real-world applications [23]. Thus, how to achieve humanlike driving with driver characteristics in longitudinal and lateral directions within the same framework while retaining the interpretability, situation awareness [24], and interaction awareness [2] of the model is the main challenge.

In this study, addressing the above issue, we aim to model individual driving behavior in both longitudinal and lateral directions within a motion planning framework for highway automated driving. The needed naturalistic human driving data of a specific driver is collected through real car experiments. Using the collected data, a motion planning method with the longitudinal and lateral characteristics of the driver, namely, a personalized motion planning method, is proposed for AVs in highway scenarios. With the proposed method, each user can obtain a personalized motion planner using his/her driving data. The considered driver characteristics are the relations between the longitudinal and lateral intentions and the faced situation. The motion planning method is based on the framework of the Baidu Apollo EM motion planner, which has a perception–planning–control hierarchy. The motion planning is decoupled into path optimization and speed optimization. For modeling driver behavior in the longitudinal direction, a car-following model is developed using the driving data. Then, the car-following model is integrated into the speed optimizer based on a weight ratio hypothesis model of the objective functional and Bayesian optimization. For the lateral direction, a Bayesian network (BN), which considers the physical states and lateral intentions of the surrounding vehicles, is built using the driving data to generate lateral behavioral decisions. A scheduling strategy is designed to help the BN to collaborate with the original trajectory decider. Additionally, the path optimizer is also personalized based on Bayesian optimization. The human likeness of the proposed motion planner in longitudinal and lateral characteristics is evaluated in the scenarios in the collected driving data and the Next Generation Simulation (NGSIM) dataset [25]. The main contributions of this study are listed as follows.

- (1) A motion planning method with driver characteristics in longitudinal and lateral directions is proposed, while the interpretability, situation awareness, and interaction awareness of the planner are retained.
- (2) Two algorithms, which are the personalized speed decision-making algorithm and the QP speed planner customizing algorithm, are proposed to personalize the speed optimizer with the driver's desired clearance and dynamic car-following characteristics. The former integrates the quadratic desired clearance model into the speed optimizer. The latter builds a dynamic model of the weights ratio of the speed optimization objective functional based on the MLCF, Bayesian optimization, and leave-one-out cross validation, using driving data. See Section 2.4.
- (3) A BN building algorithm and a continuous variable discretizing algorithm are proposed to build the BN of the personalized reference trajectory decider based on a probability tree. For a continuous variable on the probability tree, the continuous variable discretizing algorithm decides whether to partition, where to partition, and how many intervals to partition into. See Section 2.5.

The rest of this paper is organized as follows: In Section 2, the collection of human driving data is described briefly, and the proposed personalized motion planning method is detailed. Section 3 presents the simulation results and the discussions on the proposed method. Finally, the conclusion is drawn in Section 4.

2. Methods

2.1. Framework

Motion planning for AVs is a typical nonconvex three-dimensional (3D, station, lateral, and time) optimization with complex constraints. There are two types of methods for solving the optimization problem: direct 3D optimization methods (e.g., SQP algorithm [26], CFS algorithm [27], constrained iterative LQR algorithm [28]) and path-speed decoupled methods (e.g., [29]). Direct methods perform optimization in spatiotemporal state space, while decoupled methods transform the original 3D optimization problem into two 2D optimization problems (path optimization and speed optimization) and solve them successively. The main disadvantage of direct methods is the computational complexity. On the other hand, path-speed decoupled methods significantly reduce the computational complexity of trajectory optimization.

Therefore, this study adopts the EM framework of the Baidu Apollo EM motion planner (EM planner for short) [30], a path-speed decoupled method, with personalized customizations on path optimization, speed optimization, and reference trajectory decision making, as shown in Figure 1. The data center module provides all needed information, including localization, perception, prediction, high-definition (HD) map, and route information for the motion planner. In the motion planner, the candidate reference lines along lanes are generated within traffic regulations based on the HD map and navigation information. Then, lane-level motion planning is carried out in parallel for every candidate lane to produce the lane-level best trajectory. During the lane-level motion planning, a Frenet frame is first constructed based on that lane's reference line. Then, path optimization and speed optimization, which both consist of an E-step and an M-step, are performed in the Frenet frame and the station-time coordinate system, respectively. During the path optimization, information about the ego vehicle and traffic participants is projected on the station-lateral plane (SL Projection), and a smooth path is generated through a path optimizer consisting of dynamic programming (DP) path decision and quadratic programming (QP) path planning. During the speed optimization, information about obstacles is projected on the station-time plane (ST projection), following which a personalized speed optimizer consisting of speed decision and QP path planning generates a smooth speed profile. By combining the path and speed profiles, the best trajectory for every candidate lane is obtained. Finally, the personalized reference trajectory decider selects a reference trajectory among the trajectories of candidate lanes according to the driver's preference, the cost of trajectories, and traffic regulations, and transfers it to the vehicle controller.

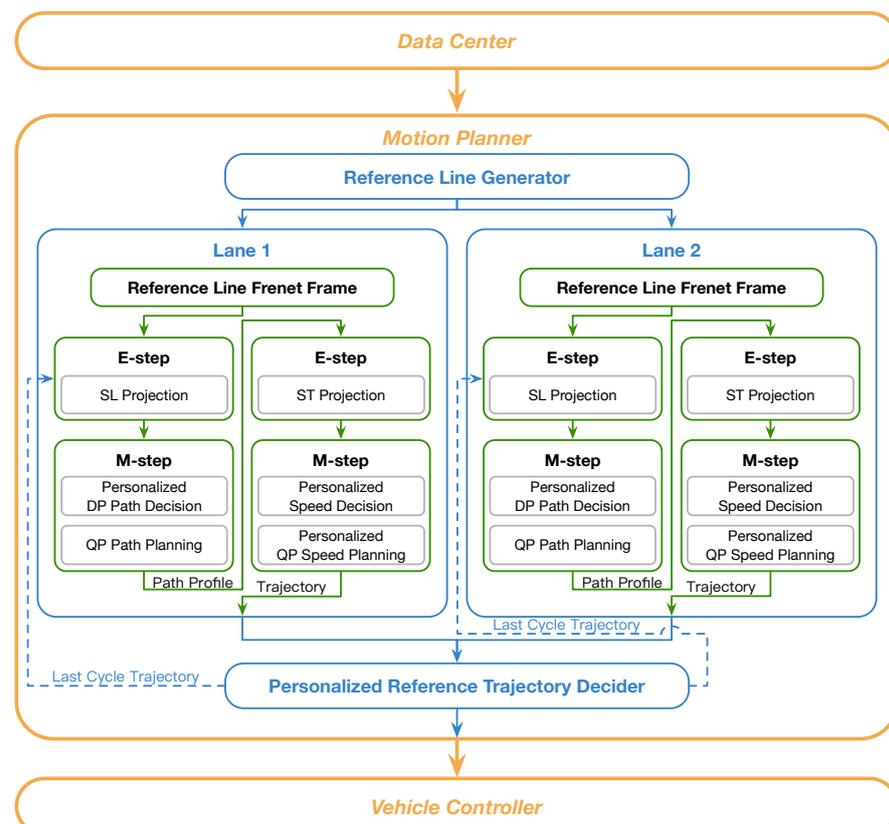


Figure 1. Framework of the motion planning method.

2.2. Human Driving Data Collection

Naturalistic human driving data are the basis for achieving humanlike driving. The routes covered in the experiment are highways in Chongqing, China. The data collecting system, embedded in the third-generation HAVAL H6, consists of a target detection and fusion computer,

a scenario acquisition system, and advanced multisource heterogeneous sensors, including an 80-channel lidar on the top, four solid lidars at the corners, front and rear radars, 360° recording cameras, etc., as shown in Figure 2. When the targets enter into the blind spot of the rotating lidar, the four corner solid lidars will provide the lidar sensor fusion module with the targets' point cloud, enhancing the reliability and stability of target detection and tracking. The image information from cameras contributes to reconstructing the traffic scenarios and estimating the lateral distance to the lane. Additionally, the inertial navigation module measures the physical states of the ego vehicle. During the data collection process, each sensor worked at a frequency of 50 Hz and transmitted data to the scenario acquisition system, where the data was saved after Kalman filtering. With this equipment, the data collection system is capable of obtaining the dimensions, category, position, velocity, and acceleration of surrounding obstacles, as well as the lateral distance to the lane, velocity, and acceleration of the ego vehicle.

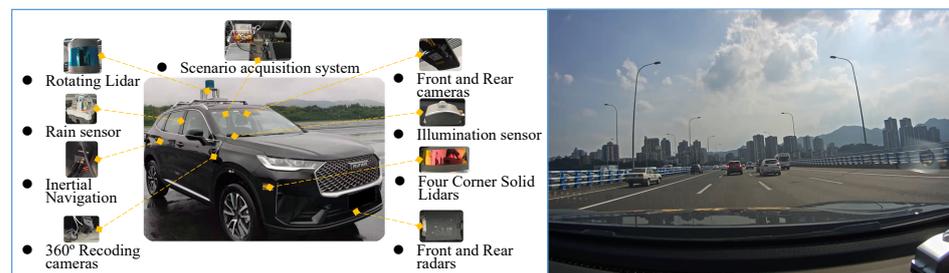


Figure 2. Human driving data collection system.

2.3. Path Optimizer

The path optimizer in this study has the same structure as the EM planner, so it is only briefly described here. For more detail, please refer to [30]. The personalization of the path optimizer will be introduced in Section 2.6. In general, the path optimization aims to obtain a smooth path represented as $l = f(s)$ for every candidate lane. In the E-step, the “Last Cycle Trajectory” (labeled in Figure 1) of the ego vehicle and the predicted trajectories of obstacles are projected on the SL plane, and the relation between the ego vehicle and the obstacles will be evaluated at each time point, as depicted in Figure 3. For motion prediction of the obstacles, the advanced methods are mainly based on machine learning, such as CNN-LSTM-based [31] and Transformer-based [32] methods, which take a sequence of historical states as the input of the deep neural networks.

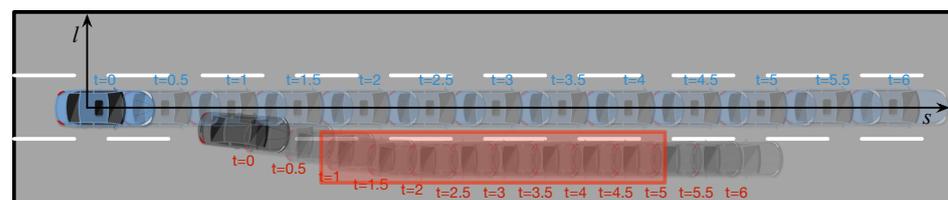


Figure 3. An example of SL projection. The red square marks the overlap region where the ego vehicle should not enter.

Once the station coordinates of the ego vehicle's bounding box intersect those of the obstacles' bounding box given a specific time, the overlap regions will be marked on the station-lateral plane, e.g., the red region in Figure 3.

In the M-step, the DP path search provides a rough path solution with a feasible tunnel, with which the spline-based QP path planning generates a smooth path. Specifically, during the DP path search, candidate paths for a specific lane are generated piecewise based on quintic polynomials. Figure 4 depicts the candidate paths generated in parallel for each lane.

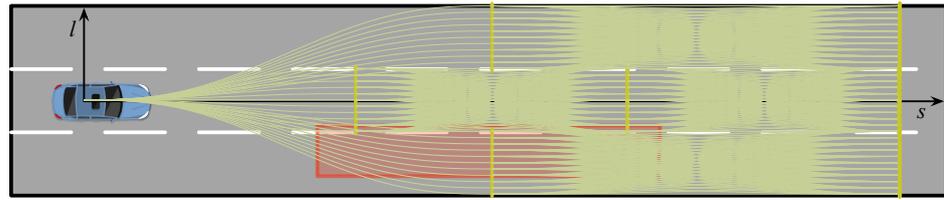


Figure 4. The candidate paths for the three lanes. The curves in olive represent the generated candidate paths for the three lanes. The white dashed lines are the boundaries between lanes. The solid black lines with arrows represent the axes of the coordinate system. The red square marks the overlap region.

Then, the candidate paths are evaluated by a cost functional, which is a linear combination of smoothness, obstacle avoidance, guidance line, and road boundary cost functionals [30]:

$$C_{\text{total}}(f) = C_{\text{smooth}}(f) + C_{\text{obs}}(f) + C_{\text{guidance}}(f) + C_{\text{boundary}}(f). \tag{1}$$

Here, the smoothness functional of a given path is defined as

$$C_{\text{smooth}}(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds + w_3 \int (f'''(s))^2 ds. \tag{2}$$

where $f'(s)$, $f''(s)$, and $f'''(s)$ correspond to the heading difference between the lane and the ego vehicle, a term related to the curvature of the path, and a term associated with the derivative of the curvature, respectively. The obstacle cost functional is defined as

$$C_{\text{obs}}(f) = \begin{cases} 0, & d_i > d_{nr} \\ \sum_0^N C_{\text{nudge}}(d_i - d_c) & d_c \leq d_i \leq d_{nr} \\ \sum_0^N C_{\text{collision}} & d_i < d_c \end{cases} \tag{3}$$

where d_i denotes the bounding box distance between the obstacle and ego vehicle at station s_i in a sequence of fixed station coordinates $\{s_0, s_1, \dots, s_N\}$. C_{nudge} is a monotonically decreasing function. d_c is the safety distance, while d_{nr} is the nudge range. $C_{\text{collision}}$ denotes a high collision cost. The guidance line cost is measured as

$$C_{\text{guidance}}(f) = \int (f(s) - g(s))^2 ds \tag{4}$$

to lead the vehicle to follow the centerline of the lane $g(s)$. The road boundary cost is set to a high value if the bounding box of the ego vehicle intersects the road boundary. By applying the DP algorithm, a candidate path with the lowest cost can be obtained, named the DP path.

With the DP path, a feasible tunnel for QP path planning is generated. The spline-based QP path planning aims to further smooth the DP path within the feasible tunnel. To reserve the smoothness and flexibility of the path, the EM planner adopts the smoothing spline function:

$$f(s) = \begin{cases} f_0(s - s_0) & s \in [s_0, s_1) \\ f_1(s - s_1) & s \in [s_1, s_2) \\ \dots & \dots \\ f_{n-1}(s - s_{n-1}) & s \in [s_{n-1}, s_n] \end{cases} \tag{5}$$

where $f_k(s) = p_{k0} + p_{k1}s + \dots + p_{km}s^m$, $k = 0, 1, \dots, n - 1$, is an m -th degree polynomial function with the coefficient $\mathbf{p}_k = (p_{k0}, p_{k1}, \dots, p_{km})^T$, and $\{s_0, s_1, \dots, s_n\}$ are the spline knots. The parameter vector of the smoothing spline function is denoted as $\mathbf{p} = (\mathbf{p}_0^T, \mathbf{p}_1^T, \dots, \mathbf{p}_{n-1}^T)^T$.

The QP path planning attempts to find a smoothing spline function $f^*(s)$ that optimizes the quadratic objective functional:

$$\arg \min_{f \in \Omega} J(f) = \sum_{i=0}^3 w_i J_i(f) \tag{6}$$

subject to linear constraints $L(f(s)) \leq \mathbf{0}$, where $\Omega = \{f : [s_0, s_n] \rightarrow \mathbb{R} \mid f, f^{(1)}, f^{(2)}, \dots, f^{(m)} \text{ is absolutely continuous and } \int_{s_0}^{s_n} (f^{(i)})^2 ds < \infty, i = 0, 1, \dots, m\}$, and w_i are weights. In detail,

$$\begin{aligned} J_0(f) &= \int_{s_0}^{s_n} (f(s) - h(s))^2 ds, \\ J_1(f) &= \int_{s_0}^{s_n} (f'(s))^2 ds, \\ J_2(f) &= \int_{s_0}^{s_n} (f''(s))^2 ds, \\ J_3(f) &= \int_{s_0}^{s_n} (f'''(s))^2 ds, \end{aligned} \tag{7}$$

where $h(s)$ is the DP path; J_0 describes the distance to the DP path; and $J_1, J_2,$ and J_3 are related to the smoothness of f . The objective functional describes the balance between obstacle avoidance and smoothness. Additionally, the linear constraints include tunnel boundary constraints and smoothness constraints. The tunnel boundary constraints require that the bounding box of the ego vehicle is constrained within the feasible tunnel, which reduces the search space and ensures the convexity of QP. The smoothness constraints require that the polynomials of the smoothing spline are joined at spline knots with up to m -th order derivative matching.

By solving the QP problem, a smooth path is obtained, as shown in Figure 5. The solid blue line is the QP path, while the DP path is represented by the dashed blue line. The orange region depicts the feasible tunnel.

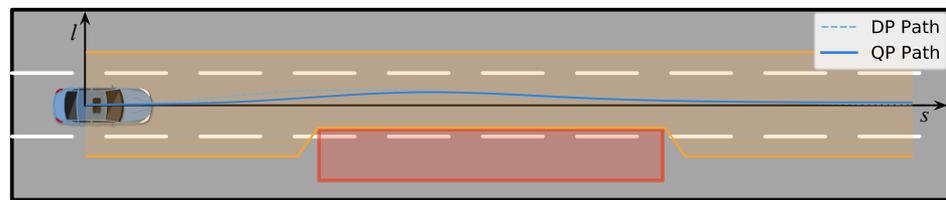


Figure 5. A QP path example of nudging. The red square marks the overlap region. The orange region depicts the feasible tunnel.

2.4. Personalized Speed Optimizer

Car following, which is one of the most common driving modes, reflects driver characteristics in the longitudinal direction. It refers to following the same preceding vehicle without changing lanes. The personalization of a speed optimizer requires the resulting longitudinal motion during car following to match the desired clearance, speed, and acceleration of a specific human driver. Therefore, the proposed personalized speed optimizer, consisting of a speed decider and a QP speed planner, is developed using the car-following segments of the collected human driving data. Since we aim to achieve humanlike driving with characteristics in both longitudinal and lateral directions in the EM framework, the goal of speed optimization is to generate a speed profile $s(t)$ (the s -coordinate of the center of the rear axle as a function of time) to be combined with the QP path $f(s)$.

There are typically three types of driver characteristics in the longitudinal direction: (1) desired clearance: a driver tries to keep a desired distance between his car and the preceding car; (2) tolerance for acceleration and deceleration: different drivers have different tolerances; and (3) dynamic car-following characteristics: the relation between the acceleration adopted by the driver and the intervehicle states (e.g., clearance, speed of the ego vehicle). In this study, the desired clearance characteristics are modeled by the

personalized speed decider, while the other two are modeled by the personalized QP speed planner.

Specifically, the desired clearance characteristics are described by the quadratic desired clearance (QDC) model [33]:

$$d_{\text{des}}(v) = av^2 + bv + c, \quad (8)$$

where d_{des} denotes the desired clearance; v is the speed of the ego vehicle; and the constant coefficients a , b , and c are identified by the least square method. Based on the QDC model, the personalized speed decision-making algorithm, as summarized in Algorithm 1, is proposed to provide a reference speed profile for personalized QP speed planning. First, with all the needed information about the ego vehicle and obstacles transferred from the data center, we calculate the predicted relative lateral distance between the ego vehicle and each obstacle. Second, the speed profiles of the preceding obstacles are extracted to mark the obstacle regions and the preliminary feasible region in the station-time plane as the red regions and the orange region shown in Figure 6, respectively. Next, given the dimensions of the ego vehicle and the minimum safety clearance (2 m in this study, less than which is considered a collision), the upper and lower bounds for a speed profile in each following segment can be calculated. The final bounds are obtained by connecting the bounds of the adjacent segments, as represented by the solid green lines in Figure 6. Finally, the personalized speed decision $s_{\text{des}}(t)$ is calculated by subtracting the desired clearance from the upper bound $s_{\text{ub}}(t)$ while satisfying $s_{\text{lb}}(t) \leq s_{\text{des}}(t) \leq s_{\text{ub}}(t)$, $s'_{\text{des}}(t) \leq v_{\text{des}}$, as represented by the dashed green line in Figure 6.

Algorithm 1: Personalized speed decision making

Input: T : planning horizon; $\{L_{\text{obs}}^i, W_{\text{obs}}^i, s_{\text{obs}}^i(t), l_{\text{obs}}^i(s) \mid i = 1, 2, \dots, N_{\text{obs}}\}$: the dimensions and predicted trajectories of the N_{obs} detected obstacles; $\{L_{\text{ego}}, W_{\text{ego}}\}$: the dimensions of the ego vehicle; $l(s)$: the QP path generated in Section 2.3; $s_{\text{last}}(t)$: last cycle speed profile; v_{des} : the desired speed set by the driver; the QDC model for the driver.

Output: $s_{\text{lb}}(t_i), s_{\text{ub}}(t_i)$: lower and upper bounds for a speed profile; $s_{\text{des}}(t)$: personalized speed decision.

- 1 Calculate the predicted relative lateral distance between the ego vehicle and each obstacle:
 $\{\Delta l^i(t) \mid \Delta l^i(t) = |l_{\text{obs}}^i(s_{\text{obs}}^i(t)) - l(s_{\text{last}}(t))|, i = 1, 2, \dots, N_{\text{obs}}, t \in [0, T]\}$;
- 2 Extract the speed profiles of the preceding obstacles
 $\{s_{\text{obs}}^i(t) \mid \Delta l^i(t) < (W_{\text{ego}} + W_{\text{obs}}^i)/2, i = 1, 2, \dots, N_{\text{obs}}, t \in [0, T]\}$, according to which the obstacle regions and the preliminary feasible region are marked in the station-time plane as the red regions and the orange region shown in Figure 6, respectively;
- 3 Calculate the upper and lower bounds, $s_{\text{ub}}(t), s_{\text{lb}}(t)$, for a speed profile in each following segment considering the dimensions of the ego vehicle and the minimum safety clearance:

$$\begin{aligned} s_{\text{ub}}(t) &= s_{\text{re}}(t) - \Delta s - Lf - Lb, \\ s_{\text{lb}}(t) &= s_{\text{fe}}(t) + \Delta s + Lr, \end{aligned}$$

where $s_{\text{re}}(t), s_{\text{fe}}(t), \Delta s$ denote the station of the rear edge of the preceding vehicle at time point t , the station of the front edge of the vehicle behind at time point t , and the minimum safety clearance, respectively. Lf, Lb , and Lr are the front overhang, wheelbase, and rear overhang of the ego vehicle, respectively;

- 4 Connect the bounds of the adjacent segments, as shown in Figure 6 by the solid green lines;
- 5 Calculate the desired stations $s_{\text{des}}(t)$ based on the QDC model:

$$\begin{aligned} s_{\text{des}}(t) &= s_{\text{ub}}(t) - d_{\text{des}}(v(t + \Delta t)), \\ \text{s.t. } s_{\text{lb}}(t) &\leq s_{\text{des}}(t) \leq s_{\text{ub}}(t), s'_{\text{des}}(t) \leq v_{\text{des}}, \end{aligned}$$

where $v(t)$ is the planned speed profile of the ego vehicle from the last planning cycle, and Δt is the period of motion planning, as shown in Figure 6 by the dashed green line.

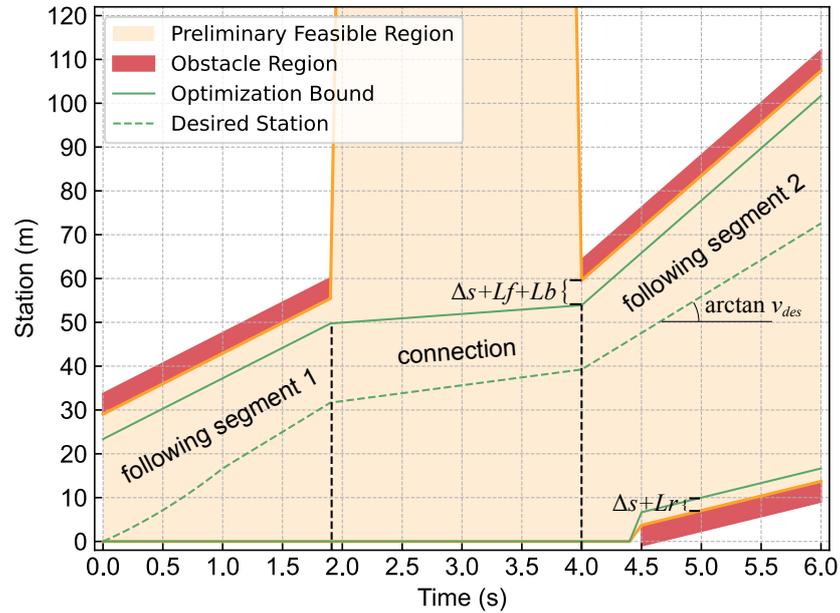


Figure 6. Personalized speed decision.

After the personalized speed decision making, a speed profile with the desired clearance characteristics of the driver is obtained. As for the tolerance for acceleration and deceleration, although constraining the $s''(t)$ and $s'''(t)$ is a direct and effective way [34], it is not considered in this study because the needed acceleration and deceleration under extreme conditions may not meet the driver’s preference. However, modeling the dynamic car-following characteristics is more difficult. Personalized speed profiles should possess the driver’s characteristics in the longitudinal direction and smoothness. Considering that the MLCF model not only mimics drivers’ longitudinal behaviors well but also has low computational cost [9,35], a personalized QP speed planning method based on the MLCF model is proposed to generate personalized smooth speed profiles. Mathematically, the QP speed planning aims to find a smoothing spline $s(t)$ defined as Equation (5) that minimizes the quadratic objective functional that has the same form as Equation (6) as well as the domain Ω of parameter vector p , subject to linear constraints $L(s(t)) \leq 0$. Here,

$$\begin{aligned}
 J_0(s) &= \int_{t_0}^{t_n} (s(t) - s_{des}(t))^2 dt, \\
 J_1(s) &= 0, \\
 J_2(s) &= \int_{t_0}^{t_n} (s''(t))^2 dt, \\
 J_3(s) &= \int_{t_0}^{t_n} (s'''(t))^2 dt,
 \end{aligned}
 \tag{9}$$

where $s_{des}(t)$ is the personalized speed decision, J_0 describes the distance to the desired station, and J_2 , and J_3 are related to the smoothness of $s(t)$. Note that $s'(t)$ is related to speed, which should not be minimized, so $J_1(s) = 0$. The linear constraints consist of boundary constraints and smoothness constraints. The boundary constraints include

$$\begin{aligned}
 s(t_i) &\leq s(t_{i+1}), i = 0, 1, 2, \dots, n - 1, \\
 s_{lb}(t_i) &\leq s(t_i) \leq s_{ub}(t_i), \\
 s'(t_i) &\leq v_{max}, \\
 a_{min} &\leq s''(t_i) \leq a_{max}, \\
 j_{min} &\leq s'''(t_i) \leq j_{max},
 \end{aligned}
 \tag{10}$$

where the first constraint forbids the vehicle from reversing, and $s_{lb}(t_i)$ and $s_{ub}(t_i)$ are the lower and upper bounds for a speed profile obtained through the personalized speed decision making. The last three constraints are related to traffic regulations and vehicle dynamics constraints, where v_{max} , a_{min} , a_{max} , j_{min} , and j_{max} are the corresponding bounds for speed, acceleration, and jerk. The determination of the above bounds involves analysis of vehicle dynamics under various scenarios. For simplicity, empirical values, $v_{max} = 33.33$ m/s, $a_{min} = -5$ m/s², $a_{max} = 5$ m/s², $j_{min} = -6$ m/s³, and $j_{max} = 6$ m/s³, were employed in this study since we focus on the topic of achieving humanlike driving. Note that driver characteristics are not considered in the last three constraints in Equation (10) because, for safety reasons, the speed, acceleration, or jerk under extreme conditions may not meet the driver’s preference. An example of the optimal solution $s^*(t)$, which balances the driver’s preference for clearance and the smoothness of the speed profile, is depicted in Figure 7.

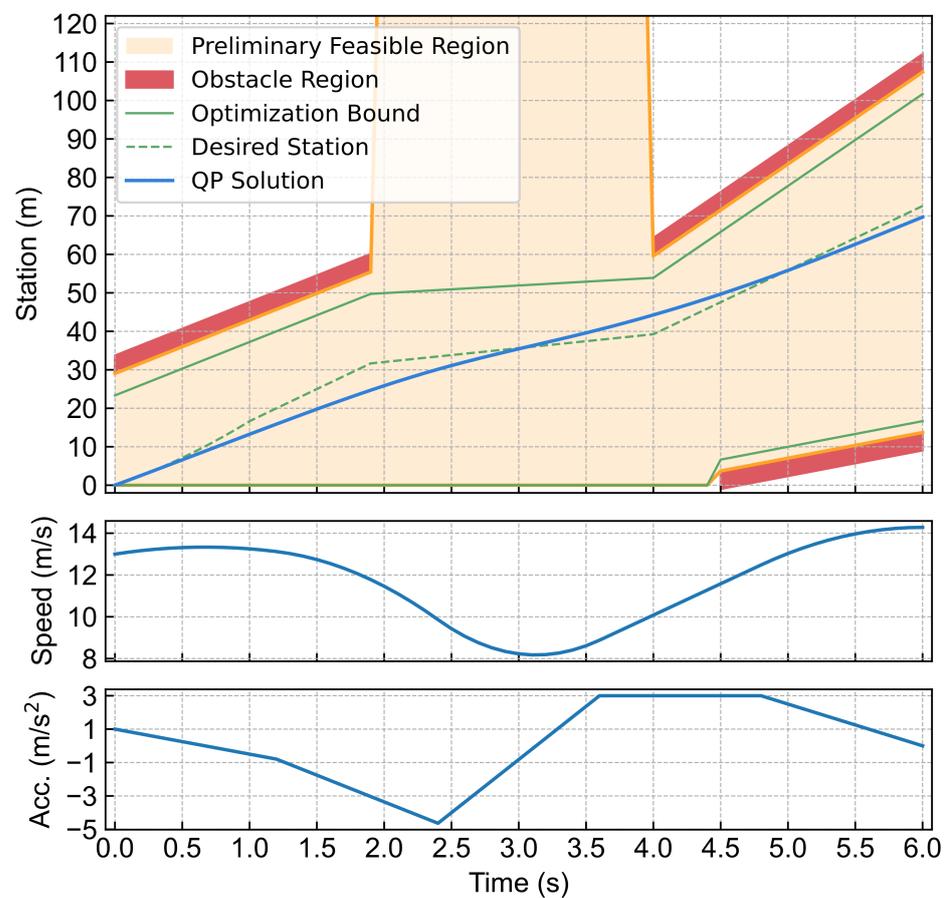


Figure 7. QP speed planning result.

Given the personalized speed decision $s_{des}(t)$ and the objective functionals in Equation (9), it is intuitive that the first objective functional $J_0(s)$ penalizes the difference between the planned station $s(t)$ and the desired station $s_{des}(t)$, while the last two terms, $J_2(s)$ and $J_3(s)$, penalize the acceleration and jerk, respectively. Meanwhile, the closer the planned station is to the desired station, the greater the acceleration is [30]. The balance between them, which is adjusted by the corresponding weights w_0 and w_2 , is actually related to the dynamic car-following characteristics. Therefore, we further explored the relation between the two weights and the resulting clearance, speed, and acceleration in car-following scenarios. Simulation results suggest that the clearance, speed, and acceleration of the resulting longitudinal motion are strongly affected by the ratio between the weights of distance to the desired station and acceleration, i.e., w_0/w_2 , as shown in Figure 8. Additionally, since $J_3(s)$ concerns the jerk,

which is not involved in the considered driver characteristics, w_3 is set to the same value as w_0 . In addition, w_1 can be set to an arbitrary value since $J_1(s) = 0$. Thus, the behavior modeling becomes finding a dynamic ratio $r = w_0/w_2$ that best describes the driver’s dynamic car-following characteristics. Then $w_2 = w_0/r$ can be easily obtained. To solve the problem, a QP speed planner customizing algorithm based on Bayesian optimization and leave-one-out cross validation (LOOCV) is proposed.

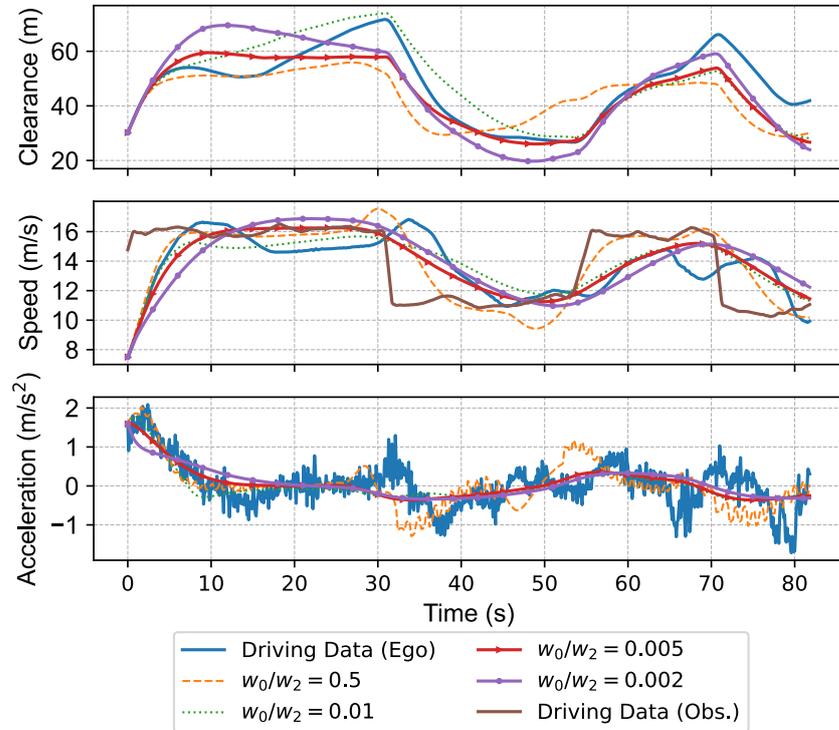


Figure 8. Comparison of clearance, speed, and acceleration between different ratios.

Considering the outstanding performance of the MLCF model, the task is transformed into establishing the relation between ratio r and acceleration a , where a is calculated by the MLCF model:

$$\begin{aligned}
 a_{MLCF} &= SVE \cdot k_v \cdot (v_p - v) + SDE \cdot k_d \cdot (d - d_{des}), \\
 SVE^{-1} &= k_{SVE} \cdot v + b_{SVE}, \\
 SDE^{-1} &= k_{SDE} \cdot v + b_{SDE}.
 \end{aligned}
 \tag{11}$$

Here, SVE and SDE are sensitivity to velocity error and sensitivity to distance error, respectively, whose reciprocal are linear functions of ego vehicle speed v . v_p is the speed of the preceding vehicle. d denotes the clearance. Constants k_{SVE} , b_{SVE} , k_{SDE} , and b_{SDE} are estimated using the driving data. Constants k_v and k_d are identified by optimizing the error between the output of the following system and the measured a , v , and d . More details of the MLCF model are introduced in [9]. Additionally, one can use other car-following models as substitutes for the MLCF model, leading to different performances of the personalized speed optimizer.

In addition, the acceleration of the planned longitudinal motion monotonously increases as r increases due to the nature of the smoothing spline [30]. Therefore, r is assumed to be linear, quadratic, and logarithmic functions of $|a_{MLCF}|$ successively, whose parameters can be obtained by Bayesian optimization [36]. Since the amount of collected data is small, LOOCV is employed to evaluate the hypothesis models. The proposed algorithm for obtaining r is summarized in Algorithm 2, where N_D denotes the number of episodes of driving data, and $N_f = N_D$ denotes the number of folds of LOOCV. Each episode

ζ_i contains the trajectories of the ego vehicle and obstacles, and the car-following episode refers to the episode in which the driver of the ego vehicle performs no other behavior except car following.

First, the car-following episodes in the human driving data are divided into training set $\{\mathcal{D}_{tr}^i\}_{i=1}^{N_f}$, validation set $\{\mathcal{D}_{va}^i\}_{i=1}^{N_f}$, and testing set $\{\mathcal{D}_{te}\}$. Then, the parameters, k and b , of each hypothesis model in \mathcal{H} is identified by Bayesian optimization (from step 2 to step 19) with the training and validation sets. Specifically, SVE , SDE , k_v , and k_d for every fold of LOOCV are obtained through the method in [9]. For fitting SVE and SDE, the effective values of the velocity error and the distance error for each speed interval are calculated using the training data, respectively. With the effective values, the constants k_{SVE} , b_{SVE} , k_{SDE} , and b_{SDE} in Equation (11) can be identified through the least square method. Additionally, k_v and k_d can be obtained by optimizing the error between the simulated trajectories of the MLCF model and the trajectories in \mathcal{D}_{tr}^i .

Next, the parameters of the hypothesis model is selected either randomly or according to the expected improvement (EI) acquisition function a_{EI} [37]:

$$a_{EI}(\mathbf{x}; \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o}) = \sigma(\mathbf{x}; \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o}) \frac{\Phi(\gamma(\mathbf{x})) + \mathcal{N}(\gamma(\mathbf{x}); 0, 1)}{(\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \mathcal{N}(\gamma(\mathbf{x}); 0, 1))}, \tag{12}$$

where $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o}$ denotes the observations consisting of parameters and the corresponding losses, N_o is the number of observations, $\sigma^2(\mathbf{x}; \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o})$ denotes the predictive variance function of the Gaussian process prior, and Φ denotes the cumulative distribution function of the standard normal distribution \mathcal{N} . γ is defined as

$$\gamma(\mathbf{x}) = \frac{\min(\{\mathbf{y}_n\}_{n=1}^{N_o}) - \mu(\mathbf{x}; \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o})}{\sigma(\mathbf{x}; \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_o})}, \tag{13}$$

where μ is the predictive mean function of the Gaussian process prior. During each iteration of Bayesian optimization, SVE , SDE , k_v , k_d , k , and b are determined, with which the QP speed planner is employed to perform car following in every episode in \mathcal{D}_{va}^j . After that, the error E_r^i is calculated, which is the evaluation of the human likeness in the longitudinal direction of the parameters in the iteration. The three terms of $e_r^{i,j}$ describe the difference between the simulated trajectories and the trajectories in \mathcal{D}_{va}^j in acceleration, speed, and clearance. The weights ω_1 , ω_2 , and ω_3 are set to 0.9, 0.09, and 0.01 by trial. With the error E_r^i , the observation set is updated by adding the current parameter vector and the error. After the Bayesian optimization, the parameters of the hypothesis models are obtained. Finally, E_r is used to evaluate the hypothesis models on the testing set, and the best one is selected to be integrated with the QP speed planner. In this way, a QP speed optimizer with the driver characteristics in the longitudinal direction is developed.

2.5. Personalized Reference Trajectory Decider

After the path optimization and the speed optimization, a trajectory with a personalized speed profile for each candidate lane is obtained, among which a reference trajectory should be selected to be transferred to the vehicle controller. In fact, different drivers facing the same situation may make completely different decisions. For example, a driver may prefer to follow the preceding vehicle, while some drivers would overtake it. Our goal is to develop a reference trajectory decider that can make the same decisions as a specific driver, namely, a personalized reference trajectory decider. The main challenge of modeling such lateral behavioral preferences is the complexity and diversity of traffic scenes.

Algorithm 2: QP speed planner customizing based on Bayesian optimization and LOOCV using human driving data

Input: $\mathcal{D} = \{\zeta_i\}_{i=1}^{N_D}$: human driving data; $\mathcal{H} = \{r \mid r(|a_{MLCF}|) = k \cdot |a_{MLCF}| + b, r(|a_{MLCF}|) = k \cdot |a_{MLCF}|^2 + b, r(|a_{MLCF}|) = k \cdot \ln(|a_{MLCF}| + 1) + b\}$: hypothesis models; N_{opt} : the maximum number of Bayesian optimization iterations; the personalized speed decider; the QP speed planner.

Output: r^*, x^* : The best hypothesis model and its parameters.

- 1 Divide the car-following episodes in \mathcal{D} into $\{\mathcal{D}_{tr}^i\}_{i=1}^{N_f}$, $\{\mathcal{D}_{va}^i\}_{i=1}^{N_f}$, and $\{\mathcal{D}_{te}\}$;
- 2 **foreach** r in \mathcal{H} **do**
- 3 Fit $\{SVE^i, SDE^i\}_{i=1}^{N_f}$ and identify $\{k_v, k_d\}_{i=1}^{N_f}$ with $\{\mathcal{D}_{tr}^i\}_{i=1}^{N_f}$ through the method in [9];
- 4 **while** $i \leq N_{opt}$ **do**
- 5 **if** $i \leq 5$ **then**
- 6 Randomly select a parameter vector $x = [k_i, b_i]^T$;
- 7 **else**
- 8 Select a parameter vector $x = [k_i, b_i]^T$ according to: $x_{next} = \arg \max_x a_{EI}$
- 9 **end**
- 10 **while** $j \leq N_f$ **do**
- 11 Simulate each car-following episode in \mathcal{D}_{va}^j setting $w_0/w_2 = r(|a_{MLCF}|)$ where $k = k_i$ and $b = b_i$;
- 12 Calculate the error between the simulated trajectories and the trajectories in \mathcal{D}_{va}^j :

$$E_r^{i,j} = \omega^T \cdot e_r^{i,j},$$

$$\omega = [\omega_1, \omega_2, \omega_3]^T,$$

$$e_r^{i,j} = [e_r^{i,j}(a), e_r^{i,j}(v), e_r^{i,j}(d)]^T,$$

$$e_r^{i,j}(x) := \sqrt{\frac{1}{\phi^l} \sum_{l=1}^{\phi^j} [x_r^i(l) - x(l)]^2}, x \in \{a', v', d'\};$$
- 13
- 14 **end**
- 15 $E_r^i = \frac{1}{N_f} \sum_{j=1}^{N_f} E_r^{i,j}$;
- 16 Add x and $y = E_r^i$ to $\{x_n, y_n\}_{n=1}^{N_0}$;
- 17 **end**
- 18 $i^* = \arg \min_i E_r^i$;
- 19 $x_r^* = [k_{i^*}, b_{i^*}]$;
- 20 **end**
- 21 Fit SVE, SDE and identify k_v, k_d with $\mathcal{D}_{tr}^1 \cup \mathcal{D}_{va}^1$;
- 22 **foreach** r in \mathcal{H} **do**
- 23 Simulate each episode in $\{\mathcal{D}_{te}\}$ with x_r^* ;
- 24 Calculate the error E_r ;
- 25 **end**
- 26 $r^* = \arg \min_r E_r$;
- 27 $x^* = x_{r^*}^*$.

We propose to develop the personalized reference trajectory decider based on a Bayesian network to model driver behavior in the lateral direction. There are two steps to building such a Bayesian network, including designing the structure \mathcal{G} and identifying the parameters θ of the network. In general, determining the structures of BNs is intractable. Fortunately, the BN structure of the proposed model can be determined by analyzing the

reasoning patterns of human drivers. When driving on a highway, a driver needs to observe the physical states and lane-changing intentions of the surrounding vehicles, as well as the physical states of the ego vehicle, to make lateral behavioral decisions. Given the information about vehicles from sensors and V2X modules, the variables can be assumed to be context-specifically independent [38]. Thus, the structure of the BN (represented by \mathcal{G}^*) is designed as depicted in Figure 9, where X_1 ($\Omega_{X_1} = \{\text{'yes'}, \text{'no'}\}$) denotes whether the ego vehicle is traveling slower than expected; X_2 ($\Omega_{X_2} = \{1, 2, \dots, N_{\text{lane}}\}$) denotes the lane the ego vehicle belongs to; X_3 ($\Omega_{X_3} = \{\text{'yes'}, \text{'no'}\}$) denotes whether the adjacent lane on the right side exists; $(X_4, X_5, X_6, X_7), \dots, (X_{n-4}, X_{n-3}, X_{n-2}, X_{n-1})$ are the physical states (including the position, speed, and acceleration in s direction relative to the ego vehicle for each surrounding vehicle) and the lateral intentions ($\{\text{'left'}, \text{'keep'}, \text{'right'}\}$, representing changing to the left lane, lane-keeping, and changing to the right lane) of the surrounding vehicles; and X_n ($\Omega_{X_n} = \{\text{'left'}, \text{'keep'}, \text{'right'}\}$) is the lateral behavioral decisions of the ego vehicle. Here, Ω_{X_i} denotes the set of possible values of a random variable X_i . Lowercase letter x_i refers to a value of a random variable X_i .

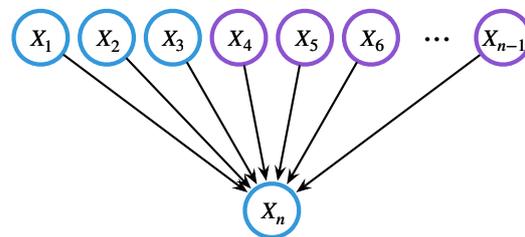


Figure 9. The BN of the personalized reference trajectory decider.

The lane-changing intentions of the surrounding vehicles and the ego vehicle can be obtained through unsupervised techniques, e.g., HDP-HSMM [39]. Note that a variable denoting the existence of the adjacent lane on the left side is not necessary. The lanes are numbered from left to right as $1, 2, \dots, N_{\text{lane}}$, and whether the adjacent lane on the left side exists is naturally known. For example, the lane on the left side does not exist if the ego vehicle drives in lane 1, while it exists if the ego vehicle drives in lane 2. Given the values of the parent variables of X_n , the BN returns the distribution of the lateral behavioral decisions of the ego vehicle.

After determining the structure, the next step is to identify the parameters θ with human driving data, i.e., the probabilities of the distribution $p(X_n | pa(X_n))$, where $pa(X_n)$ denotes the parent variable set of X_n . Since the physical states of the surrounding vehicles are continuous and other variables are discrete, the BN is a hybrid Bayesian network (HBN). Building an HBN is generally much more complex than building a BN [40]. However, considering that the output variable X_n is discrete, the HBN in this study can be transformed into a BN with only discrete variables by discretizing the continuous variables. There are many techniques for discretization, which can be divided into unsupervised (e.g., equal width and equal frequency [41]) and supervised (e.g., CAIM [42] and CACC [43]) groups. Among them, unsupervised techniques should be considered in this study because X_4, X_5, \dots, X_{n-1} are unlabeled. Meanwhile, the existing unsupervised discretizing methods make the variable space extremely large when applied to build the BN, which increases the complexity of the model because they do not utilize the underlying relationships (context-specific independence) between variables. For example, when the blue vehicle makes the lane-changing decision, it does not need to consider the vehicles in lane 1, as shown in Figure 10.

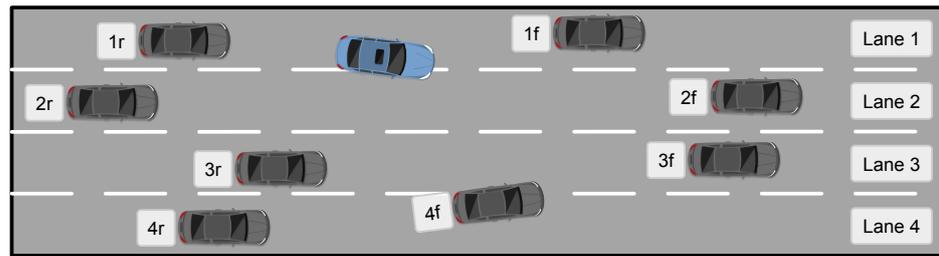


Figure 10. An example of traffic scenario.

To identify the parameters θ of the BN with the structure \mathcal{G}^* in an efficient and lightweight way, we propose an algorithm based on the probability tree [44], which discretizes all the continuous variables in the BN, as summarized in Algorithm 3. A probability tree is a directed labeled tree whose inner and leaf nodes represent random variables and probability values, respectively. The branches of an inner node represent the possible values or intervals of the variable. The path of a particular node X_i consists of the branches from the root node to it, denoted as $path(X_i)$. A sample in \mathcal{D}' is described as on the $path(X_i)$ when the values of the variables in the sample match the branches of the $path(X_i)$. In addition, during the generation of a probability tree, all variables are expanded successively, and each variable is expanded on all branches generated by the last expanded variable, as shown in Figure 11.

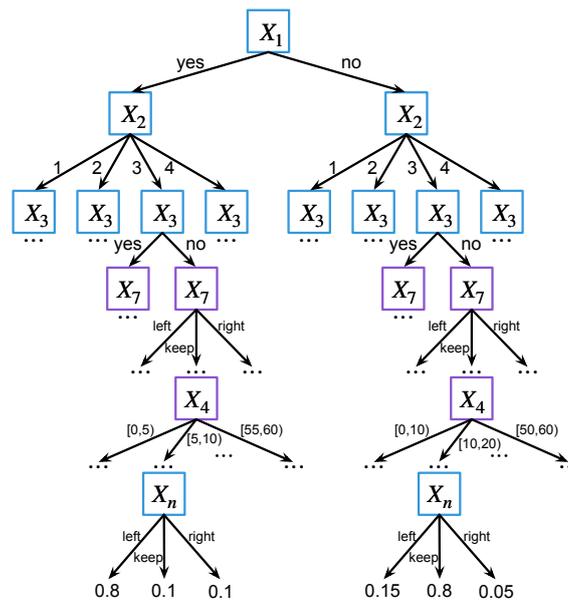


Figure 11. An example of the probability tree.

For building a probability tree, three core tasks need to be completed: (1) determining the discrete and continuous variables to be expanded, (2) determining the expanding order of the variables, and (3) determining where to partition and how many intervals to partition into for continuous variables. The proposed algorithm expands discrete variables preferentially in general. X_1 is first expanded because whether the ego vehicle reaches the driver’s desired speed may influence the driver’s lateral decision in any traffic situation, as depicted in Figure 11.

Algorithm 3: Building the BN of the personalized reference trajectory decider

Input: $\mathcal{D} = \{\zeta_i\}_{i=1}^{N_{D'}}$: human driving data; $\mathcal{N}(\mathcal{G}^*, \theta)$: the BN with the determined structure \mathcal{G}^* and an undetermined parameter vector θ .
Output: A BN $\mathcal{N}(\mathcal{G}^*, \theta^*)$ built with \mathcal{D} .

- 1 Transform \mathcal{D} into $\mathcal{D}' = \{x_i^j \mid i = 1, 2, \dots, n, j = 1, 2, \dots, N_{D'}\}$;
- 2 Expand X_1 , X_2 , and X_3 successively;
- 3 Identify the interested surrounding vehicles (ISVs) for each branch of the probability tree according to the ego lane;
- 4 Expand the lateral intention variables of the ISVs successively;
- 5 Let \mathcal{B} be the set of all branches generated by last expansion;
- 6 Let $\mathcal{C}^{\mathcal{B}}$ be the set of the physical state variables of the ISVs for each branch B in \mathcal{B} : $\{S_i^{\mathcal{B}} \mid i = 1, 2, \dots, 2N_{ISVs}\}$;
- 7 **foreach** B **in** \mathcal{B} **do**
- 8 | Obtain \mathcal{B}_B by executing Algorithm 4;
- 9 **end**
- 10 $\mathcal{T} = (\mathcal{B}, \{\mathcal{B}_B \mid B \in \mathcal{B}\})$;
- 11 Calculate the probability at each leaf of the probability tree \mathcal{T} with \mathcal{D}' based on the maximum likelihood estimation [45]: $\theta^* = p(x_n \mid \text{path}(X_n)) = \frac{\text{frequency of } X_n = x_n \text{ and } \text{pa}(X_n) = \text{path}(X_n) \text{ in } \mathcal{D}'}{\text{frequency of } \text{pa}(X_n) = \text{path}(X_n) \text{ in } \mathcal{D}'}$.

Algorithm 4: Continuous variable discretizing

Input: \mathcal{D}' ; $\mathcal{C}^{\mathcal{B}}$; n_{INR}^{\max} : the maximum number of intervals for a continuous variable to be partitioned.
Output: \mathcal{B}_B : branches that originate from B .

- 1 Let \mathcal{B}^k be the set of new branches that grow out of the branches in \mathcal{B}^{k-1} ;
- 2 Initialize $k \leftarrow 1$;
- 3 Initialize $\mathcal{B}^0 \leftarrow \{B\}$;
- 4 **foreach** $S_i^{\mathcal{B}}$ **in** $\mathcal{C}^{\mathcal{B}}$ **do**
- 5 | **foreach** b **in** \mathcal{B}^{k-1} **do**
- 6 | Find the minimum and maximum values of $S_i^{\mathcal{B}}$: $s_i^{b,\min}$ and $s_i^{b,\max}$;
- 7 | $n_{INR} = 1$;
- 8 | **while** $n_{INR} \leq n_{INR}^{\max}$ **do**
- 9 | Equally partition $[s_i^{b,\min}, s_i^{b,\max}]$ in to n_{INR} intervals;
- 10 | Calculate the probabilities of the lateral intentions of the ego vehicle for each interval w in the partitioned intervals: $p_w(x) \leftarrow \frac{n_w^k}{n_b^k}, x \in X_n$;
- 11 | Calculate the maximum KL divergence: $d_{KL, n_{INR}}^{\max} \leftarrow \max d_{KL}(p \parallel q), d_{KL}(p \parallel q) = \sum_{x \in X_n} p(x) \log \frac{p(x)}{q(x)} - \epsilon$;
- 12 | $n_{INR} \leftarrow n_{INR} + 1$;
- 13 | **end**
- 14 | $n_{INR}^{*, S_i^{\mathcal{B}}} \leftarrow \arg \max_{n_{INR}} d_{KL, n_{INR}}^{\max}$;
- 15 | Equally partition $S_i^{\mathcal{B}}$ into $n_{INR}^{*, S_i^{\mathcal{B}}}$ intervals within $[s_i^{b,\min}, s_i^{b,\max}]$;
- 16 | Expand $S_i^{\mathcal{B}}$ based on the partition, and let \mathcal{B}_b^k be the new branches that grow out;
- 17 | **end**
- 18 | $\mathcal{B}^k \leftarrow \{i \mid i \in \mathcal{B}_b^k, b \in \mathcal{B}^{k-1}\}$;
- 19 | $k \leftarrow k + 1$;
- 20 | **end**
- 21 $\mathcal{B}_B \leftarrow \{i \mid i \in \mathcal{B}^k, k = 1, 2, \dots, 2N_{ISVs}\}$.

Next, X_2 and X_3 are expanded for similar reasons. Then, the lateral intentions of the interested surrounding vehicles (ISVs) are expanded successively. Here, the ISVs are determined by analyzing the traffic scenario. For simplicity, a scenario of a highway with four lanes is used for explanation, as shown in Figure 10. First, only the two closest vehicles ahead and behind in each lane (the vehicles labeled as “ $\cdot f$ ” and “ $\cdot r$ ”) need to be considered. One reason for ignoring the vehicles farther away in front is that they generally do not directly interact with the ego vehicle. Another important reason is that considering those vehicles will make the structure of the BN much more complex, which hinders its application. In addition, there is no additional threshold to exclude vehicles that are very far because it may vary from one driver to another. Algorithm 4, presented later in this paper, can automatically partition the distance variable of an obstacle vehicle according to its effect on the ego lateral intentions. In this way, the algorithm equivalently finds the distance threshold beyond which the vehicles will not affect the ego lateral intentions.

Second, the vehicle behind the ego vehicle (labeled as “ $1r$ ”) does not affect the lateral intentions of the ego vehicle. Third, vehicles beyond the width of two lanes on the left or right (labeled as “ $4f$ ” and “ $4r$ ”) are not considered. Furthermore, the vehicle in the rear of the lane next to the adjacent lane (labeled as “ $3r$ ”) does not hinder the ego vehicle.

Moreover, if some of the ISVs are absent, their lateral intentions are considered ‘keep’, and their physical states are set to particular values, e.g., 1×10^8 . Finally, the lateral intentions of the surrounding vehicles also contribute to the determination of the ISVs; e.g., in Figure 10, vehicle “2f” is not an ISV if it is changing to the right lane. In summary, the ISVs for the ego vehicle driving in each lane are listed in Table 1, where “2r: keep” means that vehicle “2r” is regarded as an ISV only if it keeps the lane. Note that the ISVs are defined concerning time, and the selection of ISVs will be updated over time. In addition, the ISV selection rules presented in Table 1 are general rules for scenarios with four traffic lanes, and rules for scenarios with more lanes can be established by similar analysis.

Table 1. ISVs for the ego vehicle driving on a highway with four lanes.

Ego Lane	ISVs
1	“1f: keep, right”, “2f: left, keep”, “2r: keep”, “3f: left”;
2	“1f: keep, right”, “1r: keep”, “2f: left, keep, right”, “3f: left, keep”, “3r: keep”, “4f: left”;
3	“1f: right”, “2f: keep, right”, “2r: keep”, “3f: left, keep, right”, “4f: left, keep”, “4r: keep”;
4	“2f: right”, “3f: keep, right”, “3r: keep”, “4f: left, keep”.

Actually, the determination of ISVs takes advantage of the context independence between variables. With the lateral intention variables of the ISVs expanded, continuous variables (physical states of the ISVs) are partitioned and expanded successively by executing Algorithm 4. In Figure 11, for example, the lateral intention variable X_7 of the first ISV is expanded, following which the lateral intention variables X_{11} , X_{15} , and X_{19} of other ISVs (represented by “...”) are expanded successively. Then the position variable X_4 of the first ISV is partitioned and expanded, and the speed and acceleration variables X_5 and X_6 of the first ISV, as well as the physical state variables of other ISVs, are partitioned and expanded successively. Since the ISVs for each branch B in \mathcal{B} vary from those in other branches, Algorithm 4 is executed sequentially.

In Algorithm 4, the first step for discretizing a variable S_i^B at a branch b is to find its local minimum and maximum values in the subset of \mathcal{D}' whose samples are on the $path(S_i^B)$, instead of global minimum and maximum values in \mathcal{D}' . This can contribute to retaining more information when variables are partitioned into intervals. For example, assuming that the ranges of a variable in \mathcal{D}' and the subset are $[0, 100]$ and $[40, 100]$, respectively, discretization with partition 2 contains more information about the local distribution than with partition 1, as shown in Figure 12.

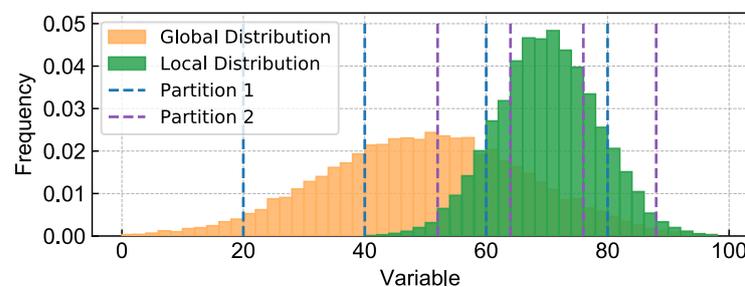


Figure 12. An example of different partitions of a variable.

Then, the number of intervals n_{INR} that the variable is partitioned into is determined according to the maximum KL divergence $d_{KL, n_{INR}}^{\max}$. Obviously, the larger the n_{INR} is, the more information about the distribution is retained. However, the growth of n_{INR} increases the

size of the probability tree, i.e., the number of the BN's parameters, exponentially, which significantly increases the time complexity and space complexity of the algorithm. Thus, $d_{KL, n_{INR}}^{\max}$ is employed to assess the gain of partitioning. The core idea is that more intervals are needed only if the increase in n_{INR} benefits maximizing the difference between the distributions of the ego lateral intentions in the partitioned intervals. Specific to a driving scenario, if the obstacle vehicle in the adjacent lane is close to the ego vehicle in the longitudinal direction, a driver would not change to that lane even though the obstacle vehicle will move away soon. In this case, there is no need to partition the speed variable of the obstacle vehicle since whether it is partitioned does not affect the lateral intentions of the driver; i.e., the distributions of the lateral intentions in the partitioned intervals are the same. To calculate $d_{KL, n_{INR}}^{\max}$, the probabilities of ego lateral intentions for each partitioned interval w are first calculated by $p_w(x) = \frac{n_b^x}{n_b}$, $x \in X_n$, where n_b denotes the number of samples that match the $path(S_i^B)$ and the interval b , and n_b^x is the number of samples in which the ego lateral intention equals $x \in \{\text{'left'}, \text{'keep'}, \text{'right'}\}$ in the samples just mentioned. Next, given two intervals, the KL divergences of them are $d_{KL}(p||q) = \sum_{x \in X_n} p(x) \log \frac{p(x)}{q(x)} - \epsilon$. Here, $p(x)$ and $q(x)$ represent the distributions of the two intervals. The KL divergences are calculated for all combinations of the intervals. The maximum KL divergence for a particular number of intervals is defined as the maximum value of all the KL divergences. Note that a small negative constant $-\epsilon$ is added to the KL divergence formula because smaller n_{INR} is preferred when the $d_{KL, n_{INR}}^{\max}$ remains the same.

By executing Algorithm 4 for each B in \mathcal{B} , all branches \mathcal{B} and $\{\mathcal{B}_B \mid B \in \mathcal{B}\}$ are obtained, according to which the probability at each leaf is estimated based on the maximum likelihood estimation (MLE) as

$$\theta^* = \frac{\text{frequency of } X_n = x_n \text{ and } pa(X_n) = path(X_n) \text{ in } \mathcal{D}'}{\text{frequency of } pa(X_n) = path(X_n) \text{ in } \mathcal{D}'}, \quad (14)$$

where $x_n \in \{\text{'left'}, \text{'keep'}, \text{'right'}\}$. At this point, a BN with the driver characteristics in the lateral direction is built. Once the information about the ego vehicle and the ISVs is given, the BN outputs a lateral decision with the highest probability based on the experience extracted from \mathcal{D}' .

However, the decisions of the BN cannot be adopted directly since the BN considers simplified information about the surrounding vehicles. On the contrary, the path optimizer and the speed optimizer plan path and speed profiles with complete information from the data center. Additionally, the lateral decisions made by the BN may be unstable during lane changing as long as a state has not been experienced in the collected human driving dataset. Therefore, a scheduling strategy must be designed to deal with the flaws. For AVs, safety is always the priority. Hence, the decision of the BN is adopted only when the corresponding trajectory is safe. The safety of a trajectory can be assessed from the kinematic aspect using the same method in the path optimizer and from the vehicle dynamics aspect via adhesion evaluation [46]. Moreover, traffic regulations should take precedence over driver's preference. As for the instability problem, theoretically, it could be solved if enough data were collected. However, it is hard to ensure the adequacy of data volume. Therefore, once a lane-changing decision is made by the BN, the ego vehicle will complete the lane-changing process unless the safety or traffic regulation requirements are not satisfied. For a situation not covered by the BN, the lateral decision will be made according to the safety assessment and traffic regulations. In addition, lateral decisions must be made after the motion planning for each candidate lane. Although making a lateral decision first and only planning motion for that decision saves computational resources, it can be dangerous in some unusual scenarios. For example, suppose a slow vehicle ahead suddenly changes lanes into the lane of the ego vehicle, while the BN decides to keep the lane. In that case, the ego vehicle may be unable to avoid the collision even with maximum deceleration. In contrast, a lane-changing decision after motion planning can be substituted for the decision of the BN and prevent the collision. With the scheduling strategy described above

and the BN, the personalized reference trajectory decider is developed, which selects a trajectory among the trajectories of candidate lanes to be transferred to the vehicle controller.

2.6. Personalizing the Path Optimizer

The personalized reference trajectory decider enables personalized lateral behavioral decision making. To further consider the lateral characteristics at the trajectory level, the path optimizer introduced in Section 2.3 needs personalization.

Remind that in the path optimizer, the DP path search provides a rough path solution, which is smoothed by the QP path planning afterward. Therefore, the DP path search should have a great impact on the generation of the final path. According to Equation (1), when the ego vehicle overlaps with obstacles or the road boundaries, C_{obs} and $C_{boundary}$ dominate the total cost C_{total} . Conversely, when the ego vehicle does not overlap with obstacles or road boundaries, C_{smooth} and $C_{guidance}$ dominate the total cost. Since safety is the priority, we can only adjust C_{smooth} and $C_{guidance}$ to improve the human likeness of the final path. The weights, w_1 , w_2 , and w_3 , in Equation (2) determine the relative magnitude of C_{smooth} and $C_{guidance}$ and thus need to be adjusted. In general, the greater the weights are, the smoother the lane-changing paths are. On the contrary, the smaller the weights are, the closer the paths are to the centerline of the lanes.

Figure 13 compares the simulated trajectories, speed, acceleration, and curvature of the ego vehicle generated by three motion planners with three sets of randomly selected weights. The motion planners had the same personalized speed optimizer and personalized reference trajectory decider. Parameter 1, Parameter 2, and Parameter 3 denote $\{w_1 = 9.30 \times 10^2, w_2 = 7.98 \times 10^4, w_3 = 9.53 \times 10^6\}$, $\{w_1 = 2.59 \times 10^2, w_2 = 6.44 \times 10^4, w_3 = 9.66 \times 10^6\}$, and $\{w_1 = 8.65 \times 10^2, w_2 = 7.12 \times 10^4, w_3 = 5.24 \times 10^6\}$, respectively. The triangles marked on the trajectories are spaced at 0.1 s intervals, representing the locations of the vehicles and indicating the traveling direction of the vehicle. The blue rectangle represents the ego vehicle, whose trajectories, speed, acceleration, and curvature in the driving data are represented by the blue curves. Those of the three motion planners compared are represented by the orange, green, and red curves. The transparent gray rectangles are the surrounding vehicles. It is shown that the weights affect not only the path but also the speed and acceleration.

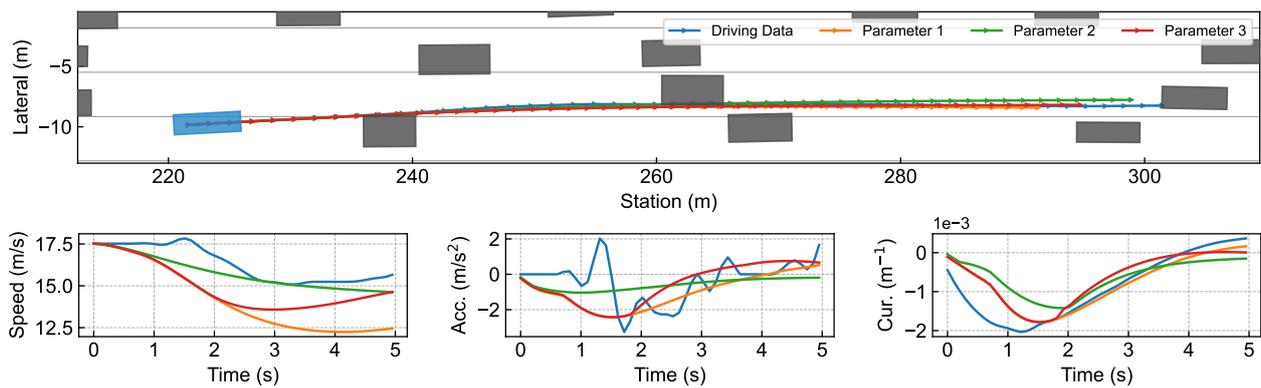


Figure 13. Comparison of trajectories, speed, acceleration, and curvature between different weights of the path optimizer.

Similar to Algorithm 2, we can also use Bayesian optimization and LOOCV to adjust the weights. The difference is that the parameter vector becomes $x = [w_1, w_2, w_3]^T$ and the optimization objective is replaced by

$$E = -\sqrt{\frac{1}{t_n + 1} \sum_{t=0}^{t_n} \|\tilde{\zeta}_{sim}(t) - \zeta(t)\|_2^2}, \tag{15}$$

where $\zeta_{\text{sim}}(t)$ stands for the simulated trajectory, and $\zeta(t)$ is the trajectory in the driving data. t_n is the number of time steps. Additionally, the personalized speed optimizer and the personalized reference trajectory decider are employed during the optimization. After the optimization, a personalized path optimizer is obtained.

3. Results and Discussion

3.1. Implementation Details

To test the proposed personalized motion planning method, we built motion planners with different models of r and with or without the BN to conduct simulations successively using the collected human driving data and the NGSIM dataset [25]. The simulations were implemented on a PC with an Intel Core i7-13700F processor (up to 5.20 GHz) and 32 GB of RAM. All the models were programmed in Python 3.9. All the simulation figures were created using the Matplotlib library 3.5.2 in Python 3.9, while the diagrams in the previous sections were created using Microsoft PowerPoint 16.68 for Mac in addition to Matplotlib. The motion planners provided a trajectory of 6 s in time length for every simulation step (every 0.1 s). The DP path consisted of two or three quintic polynomials for lane keeping or lane changing, respectively. The QP path contained five cubic polynomials with 20 parameters. During the path and speed optimization, the path or speed at 60 locations or time points was evaluated. When customizing the QP speed planner and personalizing the path optimizer, the maximum number of Bayesian optimization iterations N_{opt} was set to 100 and 1000, respectively. When building the BN, the maximum number of intervals $n_{\text{INR}}^{\text{max}}$ for a continuous variable was set to five.

The collected dataset is 1304 s long and contains both car-following and lane-changing episodes of a specific driver. For the NGSIM dataset, a segment of data from 7:50 a.m. to 8:05 a.m. on the US Highway 101 was employed. The recorded area was approximately 640 m long and consisted of five mainline lanes. The trajectory of each vehicle in the dataset is about 50 to 70 s in time length. A total of 100 vehicles that only performed car following, namely, car-following vehicles, were further selected along with their trajectories, as well as another 100 vehicles that performed lane changing, namely, lane-changing vehicles. For both datasets, 80% of the car-following episodes serve as the training and validation sets, while the rest are a testing set when customizing the QP speed planners based on Algorithm 2. Additionally, when personalizing the path optimizer, 80% of the lane-changing episodes serve as the training and validation sets, while the rest are a testing set. When building the BN, all trajectories of the lane-changing vehicles were decomposed into states at the corresponding time points. For our dataset, 80% (2863) of the states constituted the training set, while the rest (723) formed the testing set. For the NGSIM dataset, the training set and the testing set contained 96% (28631) and 4% (1192) of the states, respectively. When testing the characteristics in the lateral direction, scenarios of 10 s in time length starting from each state in the testing set were employed. The scenarios included both lane-keeping and lane-changing scenarios. By the way, the lateral behavior of the ego vehicle at each time point was used to label the lateral intention. Specifically, lane-changing time points, lane-changing start time points, and lane-changing end time points were defined as when the ego vehicle crossed the lane line, when the lateral speeds at the following five time points were all greater than 0.2 m/s, and when the lateral speeds at the following five time points were all smaller than 0.2 m/s, respectively [47]. The states between lane-changing start time and end time points were labeled with the corresponding lateral intention.

Both our dataset and the NGSIM dataset provide the positions and motion states of the ego vehicle as well as those of the surrounding vehicles, which are necessary for studying driver behavior. Note that it was assumed that the 200 drivers selected from the NGSIM dataset shared global characteristics in longitudinal and lateral directions; i.e., they had the same driving style. Additionally, we built a single personalized motion planner for the 200 drivers. This is because the proposed method is based on statistical characteristics, and the data from one vehicle are not sufficient.

In addition, during the simulation, the surrounding vehicles were assumed to follow the trajectories in the driving data, in which way all surrounding vehicles maintain humanlike behaviors. The trajectories of the surrounding vehicles were transferred to the ego vehicle as their predicted trajectories. The ego vehicle was controlled in a closed loop, planning trajectories iteratively and being assumed to track the first 0.1 s of the planned trajectory exactly in each simulation step. The kinematic state of the ego vehicle was updated according to the bicycle model in each simulation step.

3.2. Testing for the Characteristics in the Longitudinal Direction

The original EM planner and the motion planners with the constant model ($r = 0.005$), the MLCF model, and the hypothesis models of r were tested for the characteristics in the longitudinal direction in each car-following scenario in the testing set. The parameters of the hypothesis models were obtained by executing Algorithm 2, as shown in Figure 14. The colors of the heat map represent the values of error E_r^i . The blue points are sampled parameters, and the orange point is the (near) optimal parameter.

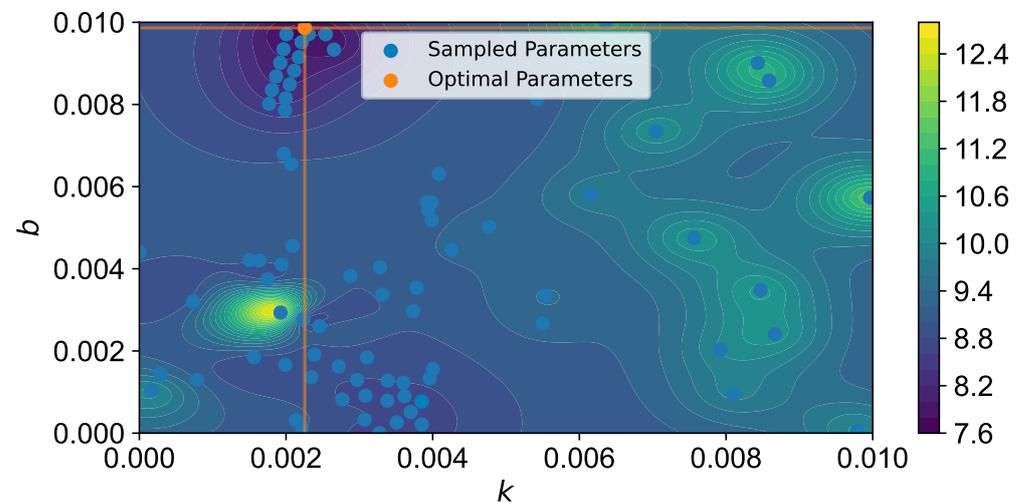


Figure 14. Bayesian optimization of the parameters of the quadratic ratio model.

The testing results on our dataset and the NGSIM dataset are given in Tables 2 and 3, respectively.

Table 2. Comparison of the EM planner, constant model, MLCF model, and hypothesis models for the human likeness in the longitudinal direction using our dataset.

Method	$\bar{e}(d)$ [m]	$\bar{e}(v)$ [m/s]	$\bar{e}(a)$ [m/s ²]	E
EM Planner	16.27	2.02	0.53	14.83
0.005	9.94	1.46	0.46	9.08
MLCF	9.24	1.24	0.45	8.43
$0.00667 a_{MLCF} + 0.00005$	9.03	1.36	0.45	8.26
$0.00226 a_{MLCF} ^2 + 0.00986$	8.48	1.30	0.46	7.75
$0.00751\ln(a_{MLCF} + 1) + 0.00779$	8.49	1.30	0.46	7.76

For each metric, the best result is in bold.

The $\bar{e}(d)$, $\bar{e}(v)$, and $\bar{e}(a)$, averaged by different episodes, denote the average errors in clearance, speed, and acceleration between the simulated trajectories and the trajectories in the driving data, respectively. The total error E is calculated as $E = 0.01\bar{e}(d) + 0.09\bar{e}(v) + 0.9\bar{e}(a)$. The smaller the total error is, the more humanlike the motion planner is. For our dataset, the results indicated that, in general, the motion planner with the quadratic ratio r outperformed other models. In contrast, the original EM planner had the worst perfor-

mance. For the NGSIM dataset, the motion planner with the quadratic ratio outperformed all other models except for the MLCF model. However, as mentioned before, the MLCF model can only perform car following, which does not meet the requirements for high-level AVs. In this sense, the quadratic ratio is still the most promising model. Figure 15 shows the details of the trajectory of a typical car-following scenario. The solid blue lines refer to the clearance, speed, and acceleration of the trajectory in the datasets, while those of the EM planner, the constant model, and the hypothesis models are represented by different lines explained in the legend. The solid pink line represents the speed of the preceding vehicle. The trajectory, whose total error E is 2.21, generated by the motion planner with the quadratic ratio is the closest to the driver’s trajectory, while the total errors of the EM planner, the constant model, the MLCF model, the linear ratio model, and the logarithmic ratio model are 6.28, 3.38, 4.12, 4.02, and 2.22, respectively.

Table 3. Comparison of the EM planner, constant model, MLCF model, and hypothesis models for the human likeness in the longitudinal direction using the NGSIM dataset.

Method	$\bar{e}(d)$ [m]	$\bar{e}(v)$ [m/s]	$\bar{e}(a)$ [m/s ²]	E
EM Planner	10.22	1.24	1.47	9.32
0.005	7.21	1.21	1.46	6.61
MLCF	5.61	1.12	1.53	5.17
$0.01872 a_{MLCF} + 0.01035$	6.35	1.04	1.44	5.82
$0.01762 a_{MLCF} ^2 + 0.01159$	6.35	1.04	1.44	5.82
$0.01973\ln(a_{MLCF} + 1) + 0.01047$	6.35	1.04	1.44	5.82

For each metric, the best result is in bold.

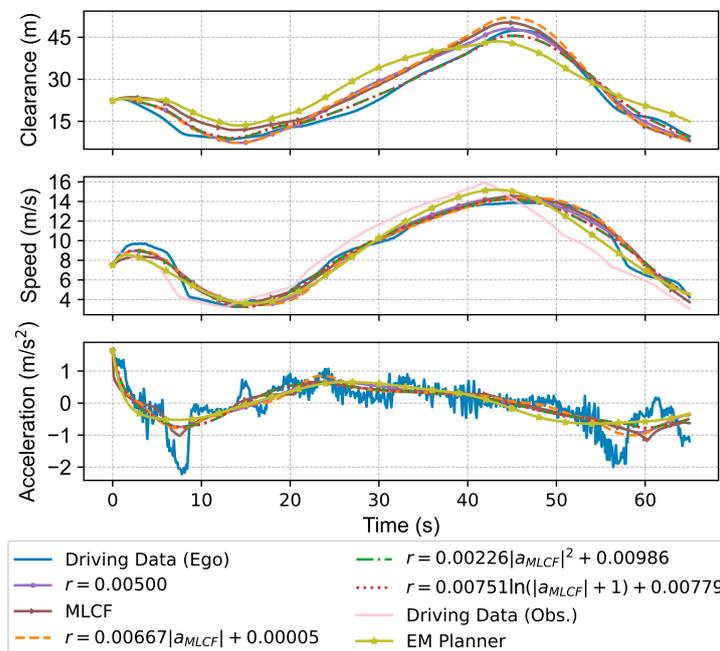


Figure 15. Comparison of the EM planner, constant model, MLCF model, and hypothesis models in a typical car-following scenario.

3.3. Testing for the Characteristics in the Lateral Direction

To test the characteristics in the lateral direction, the decisions of the BN were first evaluated. Then, the EM planner and the proposed motion planners with “PS”, “PS+PT”, and “PS+PT+PP” were employed in the testing scenarios. Here, “PS” denotes the personal-

ized speed optimizer, “PT” denotes the personalized reference trajectory decider, and “PP” denotes the personalized path optimizer.

First, the decision of the BN was compared with the lateral behavior of the ego vehicle at each time point. According to the results, for 99.86% and 95.85% of the training and the testing situations in our dataset, respectively, the BN model made the same lateral decisions as the human driver. For the NGSIM dataset, the accuracy becomes 97.30% and 92.11% for the training and the testing situations, respectively. Moreover, building a BN through the traditional method, i.e., equally partitioning the continuous variables into a particular number of intervals and estimating the probabilities based on MLE, is nearly impossible due to the vast number of parameters. For the scenarios with four lanes, as shown in Figure 10, supposing each continuous variable is partitioned into five intervals, the total number of parameters of the BN is $2 \times 4 \times 2 \times 3^8 \times 5^{16} \times 3 \approx 4.80 \times 10^{16}$. In contrast, after the determination of the ISVs, the number becomes about 3.53×10^{10} , which can be further decreased by applying Algorithm 4. The proposed algorithms for building the BN will not be compared with the traditional method since the latter is too hard to implement.

Second, the human likeness of the personalized motion planner was evaluated in the testing scenarios of our dataset and the NGSIM dataset. The human likeness is defined as the same as Equation (15)

$$HL = -\sqrt{\frac{1}{t_n + 1} \sum_{t=0}^{t_n} \|\zeta_{\text{sim}}(t) - \zeta(t)\|_2^2}, \quad (16)$$

which is similar to references [21,48]. The *HL* is the negative root-mean-square trajectory error concerning time $t = 0, 0.1, \dots, t_n$. It reflects the characteristics in longitudinal and lateral directions since it considers both longitudinal and lateral position errors at each time point. The larger the *HL* is, the more humanlike the motion planner is. The results on our dataset and the NGSIM dataset are summarized in Tables 4 and 5. All motion planners except “PS+PT+PP” used the same path optimizer with a set of hand-tuned weights $\{w_1 = 1.00 \times 10^2, w_2 = 1.00 \times 10^4, w_3 = 1.00 \times 10^6\}$. For our dataset, the weights of the personalized path optimizer in Equation (2) were $\{w_1 = 1.07 \times 10^2, w_2 = 1.14 \times 10^4, w_3 = 0.24 \times 10^6\}$, while for the NGSIM dataset, the weights were $\{w_1 = 5.13 \times 10^2, w_2 = 8.87 \times 10^4, w_3 = 9.85 \times 10^6\}$. “ \overline{HL} (all)” refers to the average human likeness of all testing scenarios, while “ \overline{HL} (lane-changing)” is the average human likeness of lane-changing scenarios in the testing set. The success rate was calculated by dividing the number of scenarios in which a motion planner generates safe trajectories by the number of scenarios in the testing set.

Table 4. Comparison of the EM planner and the proposed motion planners for the human likeness using our dataset.

Method	\overline{HL} (all) (m)	\overline{HL} (Lane Changing) (m)	Success Rate
EM planner	−7.50	−11.89	61.47%
Proposed (PS)	−4.13	−8.90	94.37%
Proposed (PS+PT)	−3.82	−5.42	96.24%
Proposed (PS+PT+PP)	−3.81	−5.32	96.71%

According to the results in Tables 4 and 5, the human likeness and success rate of the proposed motion planners were much greater than the original EM planner. Moreover, each of the proposed PS, PT, and PP is beneficial to improving the performance of the motion planner. Overall, the motion planner (PS+PT+PP) outperformed the others.

Table 5. Comparison of the EM planner and the proposed motion planners for the human likeness using the NGSIM dataset.

Method	\overline{HL} (all) (m)	\overline{HL} (Lan Changing) (m)	Success Rate
EM planner	−7.18	−10.84	64.32%
Proposed (PS)	−4.48	−6.46	97.00%
Proposed (PS+PT)	−4.49	−6.44	97.09%
Proposed (PS+PT+PP)	−4.44	−6.04	97.09%

The average computation time of a planning cycle of the proposed motion planner (PS+PT+PP) was 140 ms, while the maximum was 338 ms. As a comparison, the original EM planner took 141 ms per planning cycle on average, and the maximum was 337 ms, which suggested that our method adds almost no computational effort. Since the original EM planner has been tested in real cars, the proposed planner should also be practical in terms of computational complexity. Additionally, the acceleration and curvature of the simulated trajectories were compared with those of the trajectories in the datasets. Specifically, the average, minimum, and maximum accelerations of the trajectories generated by the proposed method were -0.067 m/s^2 , -2.413 m/s^2 , and 2.597 m/s^2 , while those in the datasets were -0.097 m/s^2 , -3.536 m/s^2 , and 3.780 m/s^2 . The average curvature and the maximum curvature of the trajectories generated by the proposed method were $4.141 \times 10^{-5} \text{ m}^{-1}$ and 0.011 m^{-1} , while those in the datasets were $9.092 \times 10^{-5} \text{ m}^{-1}$ and 0.139 m^{-1} . The absolute value of the acceleration and the curvature corresponding to the proposed method were smaller than those in the datasets. Therefore, the vehicle controller should be able to follow the planned trajectories in practice.

Figure 16 depicts the simulated trajectories, speed, acceleration, and curvature of the ego vehicle generated by the proposed motion planners in lane-changing and lane-keeping scenarios. The triangles marked on the trajectories are spaced at 0.1 s intervals, representing the locations of the vehicles and indicating the direction in which the vehicle is traveling. The blue rectangle represents the ego vehicle, whose trajectories, speed, acceleration, and curvature in the driving data are represented by the blue curves. Those of the three methods compared are represented by the orange, green, and red curves, respectively. The transparent gray rectangles are the surrounding vehicles, and the transparent gray curves depict their trajectories. In the lane-changing scenario, the human driver preferred to change to lane 2. During the lane-changing process, the vehicle in the front of lane 1 was slow and close to lane 2. Therefore, the human driver changed to lane 2 slowly. According to the simulation result, the motion planner with “PS+PT+PP” had a more similar trajectory to the human driver’s trajectory than the motion planner with “PS+PT”. Furthermore, without the personalized reference trajectory decider, the EM planner and the motion planner with only “PS” did not make the same behavioral decision as the human driver. In the lane-keeping scenario, the preceding vehicle was about to change to lane 3, and all motion planners generated lane-keeping trajectories. However, compared with the EM planner, the proposed three planners generated trajectories more similar to the human driver’s trajectory regarding the longitudinal speed.

3.4. Discussion

According to the above results, it is validated that the proposed personalized motion planning method can capture the driver characteristics in longitudinal and lateral directions using naturalistic human driving data and produce humanlike behavior within the framework of the Apollo EM motion planner. The proposed method can improve driving comfort because the resulting behavior of the AV meets the driver’s preferences. Meanwhile, the personalized motion planner is highly interpretable, situation-aware, and interaction-aware. First, interpretability is retained because the modules of the motion planner have clear mathematical structures and parameters with physical meanings. For example, the terms in the optimization objective functionals and constraints correspond to actual physical

quantities, the structure of the BN describes the causal relationship between the traffic situation and the lateral intentions, and the parameters of the BN are the probabilities of lateral intentions under different situations. Second, the personalized motion planner is situation-aware since it takes as input the locations, motion states, lateral intentions, and predicted trajectories of the surrounding vehicles. The situation awareness is modeled by the hypothesis model and the BN. Third, the mutual dependence of the ego vehicle's lateral behavioral decision and the surrounding vehicles' intentions is modeled by the BN, achieving interaction awareness of the personalized motion planner. The interpretability enables one to modify, verify, interpret, and understand the model, while situation awareness and interaction awareness can help AVs better share the road with human drivers. These properties should facilitate the practical applications of the proposed motion planning method.

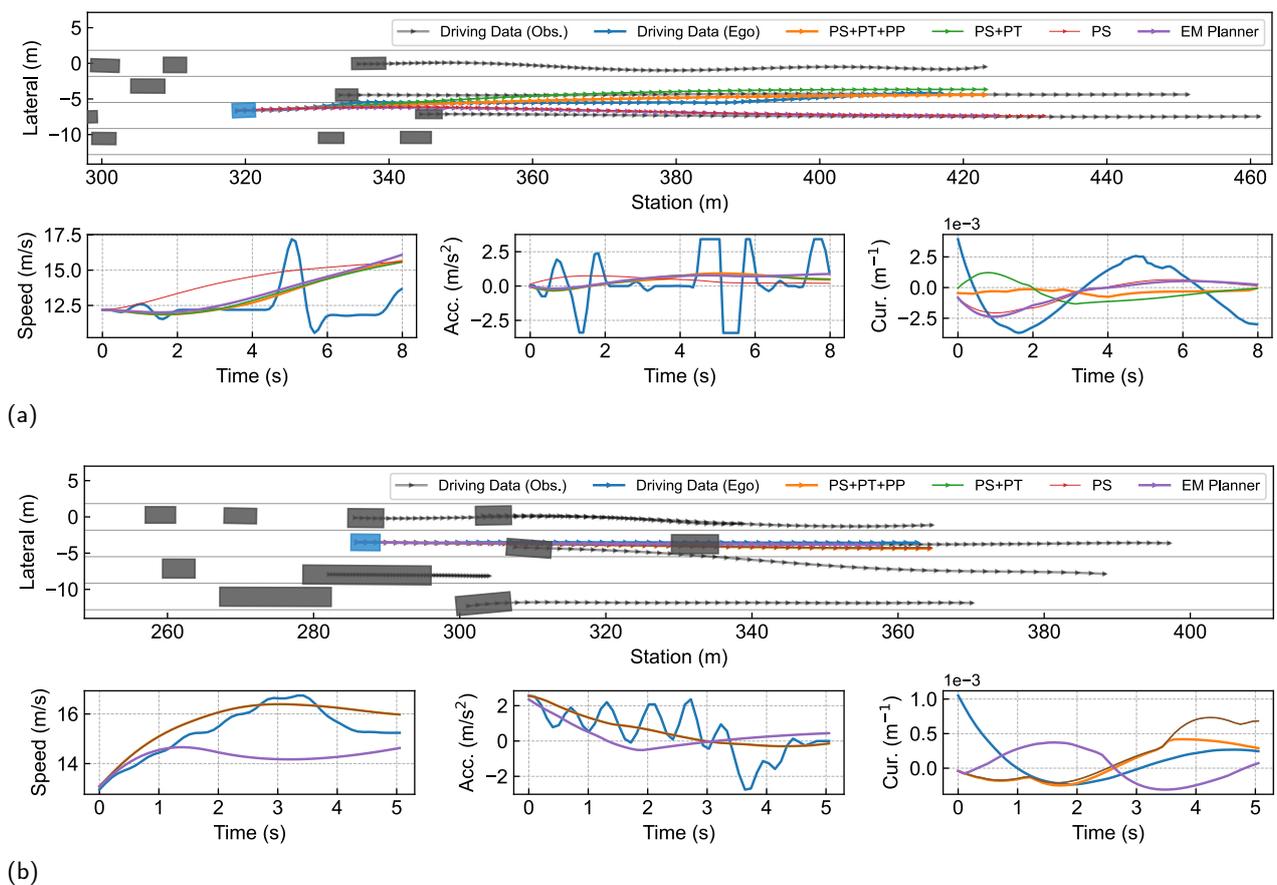


Figure 16. The simulated trajectories, speed, acceleration, and curvature of the ego vehicle in representative lane-changing and lane-keeping scenarios: (a) the lane-changing scenario; (b) the lane-keeping scenario.

Moreover, learning driver behavior also has the potential to improve driving safety. Traditional model-driven methods, such as motion planning based purely on numerical optimization, face substantial challenges in practical applications. One of the major challenges is that the driving policies of these methods are inflexible and can hardly adapt to various driving scenarios. In contrast, the proposed method enables the AVs to make humanlike decisions and thus helps the AVs with handling scenarios similar to the training scenarios.

However, the limitation of this study is that the driving behaviors in the human driving data used to build the personalized motion planner were assumed to be appropriate. Meanwhile, the behavior of drivers, especially novice drivers, may occasionally cause safety concerns or disrupt traffic flow. Although our approach imposes safety constraints on the motion planner and considers the lateral intentions of the surrounding vehicles, the human

driving data with inappropriate driving behaviors should be filtered out in advance, which is related to the field of driving behavior analysis [49]. In addition, for situations where a novice driver lacks driving data, the personalized reference trajectory decider cannot make a personalized decision. However, thanks to the interpretability of the BN, such situations can be identified when they happen because there is no corresponding probability at the leaves of the probability tree. Additionally, the decisions made by another BN built with all available driving data (including data of other drivers) should be adopted, or the decisions can be made based on trajectory safety assessment and traffic regulations. Finally, the driver's behavior can be influenced by the external conditions, e.g., icy road and fog. Benefitting from the interpretability of the proposed method, adaptive strategies can be developed to handle different external conditions. For example, the desired clearance calculated by Equation (8) can be multiplied by a factor to increase the clearance to the preceding vehicle when the road is icy. Alternatively, different personalized motion planners can be developed for different external conditions if corresponding human driving data are sufficient.

4. Conclusions

This study achieves humanlike driving with driver characteristics in longitudinal and lateral directions within the same motion planning framework. The motion planning of the ego vehicle is decoupled into path optimization and speed optimization. To introduce the driver characteristics in the longitudinal direction into the motion planner, two algorithms are proposed to personalize the speed optimizer. Specifically, the desired clearance characteristics are modeled by the QDC, which is further integrated into the speed decision-making process. Then, the weight ratio of the objective functional is assumed to be linear, quadratic, and logarithmic functions of the acceleration calculated by the MLCF, whose parameters can be obtained by Bayesian optimization and LOOCV using the driving data. To personalize the motion planner with the driver characteristics in the lateral direction, a BN is employed to model the lateral behavioral decision-making preference. Two algorithms, which partition continuous variables according to the maximum KL divergence based on a probability tree, are proposed to build the BN in an efficient and lightweight way. Moreover, a scheduling strategy is designed to weigh the BN decisions against traffic regulations and trajectory costs. Finally, a personalized motion planner within the Apollo EM motion planning framework is developed by applying the proposed algorithms using the driving data. The results of the simulation constructed based on real driving data validated the effectiveness of the proposed method and the feasibility of achieving humanlike driving with driver characteristics in longitudinal and lateral directions under the same motion planning framework. It is also indicated that the proposed Algorithm 2 is able to find the best model of ratio r among the hypothesis models. Furthermore, with the proposed Algorithms 3 and 4, the number of the parameters of the BN is significantly reduced, while the precision of the BN is preserved as much as possible, considering the limited time complexity and space complexity.

Author Contributions: Conceptualization, D.Z.; methodology, D.Z.; software, D.Z.; validation, D.Z.; formal analysis, D.Z.; investigation, D.Z.; resources, J.Z. and K.W.; data curation, J.Z. and K.W.; writing—original draft preparation, D.Z.; writing—review and editing, D.Z.; visualization, D.Z.; supervision, L.Z. and Y.L.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 51875061) and the Fundamental Research Funds for the Central Universities (2023CDJXY-021).

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of Chongqing University Cancer Hospital (CZLS2021215-A).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The datasets that support this study include a dataset generated during the study and a publicly archived dataset named the NGSIM dataset. Although our dataset cannot be shared due to project policy, the NGSIM dataset is available at <https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm> (accessed on 17 February 2022).

Conflicts of Interest: Author Jie Zeng and Kan Wang were employed by the company China Merchants Testing Vehicle Technology Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BN	Bayesian network
AVs	automated vehicles
IDM	intelligent driver model
LCF model	linear car-following model
MLCF model	modified linear car-following model
IRL	inverse reinforcement learning
PACC	personalized adaptive cruise control
MPC	model predictive control
CNN	convolutional neural network
DL	deep learning
LSTM	long short-term memory
NGSIM	Next Generation Simulation
HD map	high-definition map
SL	station-lateral
ST	station-time
DP	dynamic programming
QP	quadratic programming
QDC model	quadratic desired clearance model
LOOCV	leave-one-out cross validation
SVE	sensitivity to velocity error
SDE	sensitivity to distance error
HBN	hybrid Bayesian network
ISVs	interested surrounding vehicles
MLE	maximum likelihood estimation
PS	personalized speed optimizer
PT	personalized reference trajectory decider
PP	personalized path optimizer

References

- Li, L.; Ota, K.; Dong, M. Humanlike driving: Empirical decision-making system for autonomous vehicles. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6814–6823. [CrossRef]
- Xing, Y.; Lv, C.; Wang, H.; Wang, H.; Ai, Y.; Cao, D.; Velenis, E.; Wang, F.Y. Driver lane change intention inference for intelligent vehicles: framework, survey, and challenges. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4377–4390. [CrossRef]
- Sharma, O.; Sahoo, N.C.; Puhan, N. Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104211. [CrossRef]
- Brown, K.; Driggs-Campbell, K.; Kochenderfer, M.J. A Taxonomy and Review of Algorithms for Modeling and Predicting Human Driver Behavior. *arXiv* **2020**, arXiv:2006.08832.
- Donges, E. A conceptual framework for active safety in road traffic. *Veh. Syst. Dyn.* **1999**, *32*, 113–128. [CrossRef]
- Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805. [CrossRef]
- Wiedemann, R. Simulation des Strassenverkehrsflusses. Ph.D. Thesis, Universität Karlsruhe: Karlsruhe, Germany, 1974.
- Brackstone, M.; McDonald, M. Car-following: A historical review. *Transp. Res. Part F: Traffic Psychol. Behav.* **1999**, *2*, 181–196. [CrossRef]
- Li, S.; Wang, J.; Li, K.; Lian, X.; Ukawa, H.; Bai, D. Modeling and verification of heavy-duty truck drivers' car-following characteristics. *Int. J. Automot. Technol.* **2010**, *11*, 81–87. [CrossRef]

10. Li, S.; Li, K.; Rajamani, R.; Wang, J. Model predictive multi-objective vehicular adaptive cruise control. *IEEE Trans. Control Syst. Technol.* **2010**, *19*, 556–566. [[CrossRef](#)]
11. Ozkan, M.F.; Ma, Y. Personalized Adaptive Cruise Control and Impacts on Mixed Traffic. In Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 21–28 May 2021; pp. 412–417. [[CrossRef](#)]
12. Zhao, Z.; Wang, Z.; Han, K.; Gupta, R.; Tiwari, P.; Wu, G.; Barth, M.J. Personalized Car Following for Autonomous Driving with Inverse Reinforcement Learning. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 2891–2897. [[CrossRef](#)]
13. Naranjo, J.E.; Gonzalez, C.; Garcia, R.; De Pedro, T. Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 438–450. [[CrossRef](#)]
14. Balal, E.; Cheu, R.L.; Sarkodie-Gyan, T. A binary decision model for discretionary lane changing move based on fuzzy inference system. *Transp. Res. Part C Emerg. Technol.* **2016**, *67*, 47–61. [[CrossRef](#)]
15. Huang, C.; Huang, H.; Hang, P.; Gao, H.; Wu, J.; Huang, Z.; Lv, C. Personalized Trajectory Planning and Control of Lane-Change Maneuvers for Autonomous Driving. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5511–5523. [[CrossRef](#)]
16. Yang, W.; Zheng, L.; Li, Y.; Ren, Y.; Xiong, Z. Automated highway driving decision considering driver characteristics. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2350–2359. [[CrossRef](#)]
17. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
18. Chen, L.; Hu, X.; Tian, W.; Wang, H.; Cao, D.; Wang, F.Y. Parallel planning: A new motion planning framework for autonomous driving. *IEEE/CAA J. Autom. Sin.* **2018**, *6*, 236–246. [[CrossRef](#)]
19. Zhu, Z.; Zhao, H. A Survey of Deep RL and IL for Autonomous Driving Policy Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14043–14065. [[CrossRef](#)]
20. Rosbach, S.; James, V.; Großjohann, S.; Homoceanu, S.; Roth, S. Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 2658–2665.
21. Huang, Z.; Wu, J.; Lv, C. Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10239–10251. [[CrossRef](#)]
22. Huang, L.; Guo, H.; Zhang, R.; Wang, H.; Wu, J. Capturing Drivers' Lane Changing Behaviors on Operational Level by Data Driven Methods. *IEEE Access* **2018**, *6*, 57497–57506. [[CrossRef](#)]
23. Aradi, S. Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 740–759. [[CrossRef](#)]
24. Gugerty, L. Situation awareness in driving. *Handb. Driv. Simul. Eng. Med. Psychol.* **2011**, *1*, 265–272.
25. Alexiadis, V.; Colyar, J.; Halkias, J.; Hranac, R.; McHale, G. The next generation simulation program. *Inst. Transp. Engineers. ITE J.* **2004**, *74*, 22.
26. Lim, W.; Lee, S.; Sunwoo, M.; Jo, K. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 613–626. [[CrossRef](#)]
27. Liu, C.; Lin, C.Y.; Tomizuka, M. The convex feasible set algorithm for real time optimization in motion planning. *SIAM J. Control Optim.* **2018**, *56*, 2712–2733. [[CrossRef](#)]
28. Chen, J.; Zhan, W.; Tomizuka, M. Autonomous driving motion planning with constrained iterative LQR. *IEEE Trans. Intell. Veh.* **2019**, *4*, 244–254. [[CrossRef](#)]
29. Meng, Y.; Wu, Y.; Gu, Q.; Liu, L. A decoupled trajectory planning framework based on the integration of lattice searching and convex optimization. *IEEE Access* **2019**, *7*, 130530–130551. [[CrossRef](#)]
30. Fan, H.; Zhu, F.; Liu, C.; Zhang, L.; Zhuang, L.; Li, D.; Zhu, W.; Hu, J.; Li, H.; Kong, Q. Baidu apollo em motion planner. *arXiv* **2018**, arXiv:1807.08048.
31. Xie, G.; Shangguan, A.; Fei, R.; Ji, W.; Ma, W.; Hei, X. Motion trajectory prediction based on a CNN-LSTM sequential model. *Sci. China Inf. Sci.* **2020**, *63*, 1–21. [[CrossRef](#)]
32. Zhou, Z.; Ye, L.; Wang, J.; Wu, K.; Lu, K. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8823–8833.
33. Yi, K.; Han, D. Adaptive Cruise Control Algorithm Design Based on Human Drivers' Driving Pattern. In Proceedings of the 9th World Multi-Conference, Systems, Cybernetics and Informatics, Orlando, FL, USA, 10–13 July 2005; Volume 2, pp. 205–210.
34. Yi, K.; Moon, I.K. A driver-adaptive stop-and-go cruise control strategy. In Proceedings of the IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, 21–23 March 2004; Volume 1, pp. 601–606.
35. Li, S.E.; Li, K.; Wang, J. Economy-oriented vehicle adaptive cruise control with coordinating multiple objectives function. *Veh. Syst. Dyn.* **2013**, *51*, 1–17. [[CrossRef](#)]
36. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [[CrossRef](#)]
37. Jones, D.R.; Schonlau, M.; Welch, W.J. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [[CrossRef](#)]
38. Boutilier, C.; Friedman, N.; Goldszmidt, M.; Koller, D. Context-specific independence in Bayesian networks. *arXiv* **2013**, arXiv:1302.3562.

39. Taniguchi, T.; Nagasaka, S.; Hitomi, K.; Takenaka, K.; Bando, T. Unsupervised hierarchical modeling of driving behavior and prediction of contextual changing points. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1746–1760. [[CrossRef](#)]
40. Salmerón, A.; Rumí, R.; Langseth, H.; Nielsen, T.; Madsen, A. A Review of Inference Algorithms for Hybrid Bayesian Networks. *J. Artif. Intell. Res.* **2018**, *62*, 799–828. [[CrossRef](#)]
41. Dougherty, J.; Kohavi, R.; Sahami, M. Supervised and Unsupervised Discretization of Continuous Features. In *Machine Learning Proceedings 1995*; Friediris, A.; Russell, S., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 194–202.
42. Kurgan, L.A.; Cios, K.J. CAIM discretization algorithm. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 145–153. [[CrossRef](#)]
43. Tsai, C.J.; Lee, C.I.; Yang, W.P. A discretization algorithm based on class-attribute contingency coefficient. *Inf. Sci.* **2008**, *178*, 714–731. [[CrossRef](#)]
44. Salmerón, A.; Cano, A.; Moral, S. Importance sampling in Bayesian networks using probability trees. *Comput. Stat. Data Anal.* **2000**, *34*, 387–413. [[CrossRef](#)]
45. Koski, T.; Noble, J. *Bayesian Networks: An Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 924.
46. Pugi, L.; Favilli, T.; Berzi, L.; Locorotondo, E.; Pierini, M. Brake blending and torque vectoring of road electric vehicles: A flexible approach based on smart torque allocation. *Int. J. Electr. Hybrid Veh.* **2020**, *12*, 87–115. [[CrossRef](#)]
47. Wang, Q.; Li, Z.; Li, L. Investigation of discretionary lane-change characteristics using next-generation simulation data sets. *J. Intell. Transp. Syst.* **2014**, *18*, 246–253. [[CrossRef](#)]
48. Sun, R.; Hu, S.; Zhao, H.; Moze, M.; Aioun, F.; Guillemard, F. Human-like highway trajectory modeling based on inverse reinforcement learning. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 1482–1489.
49. Abou Elasad, Z.E.; Mousannif, H.; Al Moatassime, H.; Karkouch, A. The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103312. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.