scientific reports

OPEN

Research on vehicle trajectory planning algorithm integrating spatiotemporal constraints and adaptive curvature

Yatao Zhao✉, Jiangfeng Wang, Peikun Li & Hao Wang

The trajectory planning method based on the Frenet coordinate transformation effectively reduces computational complexity and enhances planning accuracy. However, in complex road scenarios involving multiple intelligent agents, the singular Frenet coordinate transformation is incapable of handling the variations in road curvature and vehicle posture among multiple target vehicles. This limitation leads to a reduced planning solution space, making it difficult to approximate the optimal path. To address this challenge, this study proposes a curvature-adaptive trajectory planning framework combined with spatiotemporal integration. This method involves three key innovations: (1) A road segmentation algorithm that discretizes complex curved roads into a set of quasi-straight segments through adaptive spatial discretization; (2) A multi-coordinate system adaptive transformation based on the predicted vehicle center in the temporal domain, which integrates temporal constraints with spatial Bézier curve optimization in a hierarchical planning architecture; (3) A multi-objective cost function is designed to comprehensively evaluate heuristic features such as path smoothness, efficiency, and safety, and candidate paths are optimized and ranked through weight configuration. Experimental results demonstrate that, compared with the benchmark method, the proposed framework achieves a 100% planning success rate in 135 complex road scenarios, with a 6.15% improvement in planning robustness and a 9.41% increase in trajectory smoothness. These findings confirm that the proposed framework has certain advantages in terms of smoothness and safety in complex road scenarios.

Keywords Autonomous driving, Trajectory planning, Spatiotemporal integration, Frenet coordinate system, Bézier curves

With the rapid development of autonomous driving technology, trajectory planning has become an essential core function in intelligent driving systems, playing a crucial role in improving traffic efficiency and reducing traffic accidents[1–3]. This is especially true in complex road scenarios involving multiple intelligent agents, where intelligent vehicles must not only accurately plan their trajectories but also ensure safety and efficiency during operation. However, trajectory planning in multi-agent and curved-road scenarios is challenging, as vehicles need to consider geometric constraints of the road, predict the intentions of surrounding vehicles, and make optimal decisions within limited time frames[4–6].

Traditional trajectory planning methods, often based on rule-based systems or model predictive control, perform well in simple scenarios but may fail to provide sufficient comfort and robustness in complex situations[7]. Advanced methods based on deep reinforcement learning (DRL), although capable of learning complex decision-making strategies, still face challenges in generalization and robustness[8].

Conventional decoupled spatiotemporal decision-making and planning algorithms, which separate the temporal and spatial dimensions, can ensure real-time performance but often result in conservative or aggressive trajectory planning due to the neglect of spatiotemporal correlations[9]. Moreover, single-dimensional safety mechanisms are insufficient to address sudden risks in complex interactive scenarios[10]. Therefore, achieving joint spatiotemporal optimization and multi-layered safety protection while maintaining real-time performance has become a key issue in enhancing the reliability of intelligent driving systems.

Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Ministry of Transport, Beijing Jiaotong University, Beijing 100044, China. ✉email: 21114101@bjtu.edu.cn

Joint spatiotemporal trajectory planning algorithms are among the critical technologies in intelligent driving trajectory planning. These algorithms consider both temporal and spatial dimensions to achieve globally optimal trajectory planning[11–13]. However, existing joint spatiotemporal trajectory planning algorithms fail to fully account for the effects of road curvature and vehicle posture, resulting in a limited solution space and an inability to approximate the optimal solution.

To address these challenges, this paper proposes a joint spatiotemporal planning algorithm that incorporates heuristic human-like decision-making thinking. First, the proposed curvature-adaptive joint spatiotemporal trajectory planning algorithm achieves adaptive road representation, effectively decoupling the constraints of complex road curves and vehicle posture. Subsequently, the hierarchical Bézier optimization algorithm balances robustness and spatial smoothness. Finally, by integrating planning costs with real-time control, the algorithm achieves human-like decision-making capabilities.

Related works

As a core technology in intelligent driving systems, vehicle trajectory planning is perpetually confronted with the challenge of optimal solution search under multiple constraints. Its fundamental objective is to generate an optimal motion trajectory from the initial state to the target state while satisfying the triple constraints of dynamics, kinematics, and safety conditions. Building on this foundation, numerous scholars have conducted in-depth research and proposed a variety of algorithms and strategies.

Search-based planning methods rely on the discretization of environmental space to construct a solution framework. Traditional search algorithms, represented by Dijkstra and A*[14–16], achieve path search through state grid discretization, with the advantage of strict completeness guarantees. Among them, the A* algorithm, by incorporating a heuristic function, prioritizes the expansion of nodes closer to the goal[17]. However, it still faces efficiency bottlenecks when dealing with high-dimensional state spaces. Notably, the MHA* algorithm significantly improves search efficiency through a multi-heuristic fusion mechanism, yet the trajectories it generates often neglect the kinodynamic constraints of vehicles, resulting in insufficient trajectory continuity. The spatiotemporal grid expansion method innovatively introduces obstacle motion prediction, achieving a breakthrough in trajectory planning under dynamic environments. However, this method still faces the dual challenges of exponentially increasing computational complexity and insufficient trajectory smoothness.

Sampling-based planning methods, represented by Rapidly-exploring Random Tree (RRT)[18] and Probabilistic Roadmap (PRM)[19], demonstrate unique advantages in planning within high-dimensional spaces. Their core value lies in achieving probabilistic completeness, thereby overcoming the curse of dimensionality. However, the inherent sampling uncertainty leads to significant fluctuations in path quality. In terms of technological evolution, Time-extended Batched Rapidly-exploring Random Tree (TB-RRT)[20] enhances adaptability to dynamic environments through temporal dimension expansion. Nevertheless, the stochastic sampling mechanism still suffers from inherent drawbacks, such as excessive computational resource consumption and insufficient trajectory stability. Improved RRT algorithms have made breakthroughs in sampling efficiency, but the fundamental issue of trajectory feasibility verification remains unresolved.

Parametric curve methods realize trajectory geometric constraints through preset parametric curves, with typical representatives including Bézier curves and quintic polynomials[21–23]. Among them, the innovative practice of optimizing the end-point curvature of the trajectory using a fifth-order Bézier curve has engineering value, but it exposes the shortcoming of lacking a rapid determination mechanism for control points in emergency obstacle avoidance scenarios[24]. In comparison, the piecewise polynomial decoupled planning method significantly improves trajectory continuity, but there is still room for improvement in the computational efficiency of its parameter optimization process[25].

With the advancement of intelligent algorithms, hybrid intelligent methods have demonstrated advantages in obstacle avoidance success rates. The artificial potential field method achieves efficient planning by constructing a virtual gradient field, and the design of the virtual force field has successfully eliminated trajectory oscillation[26–28]. The integration of potential field methods with optimal control theory has significantly enhanced trajectory feasibility. However, potential field modeling in complex scenarios remains an unresolved challenge. Emerging methods based on deep reinforcement learning have shown strong adaptability to environments[8]. Although deep predictive networks can capture behavioral uncertainty, their inference latency still exceeds the real-time control threshold, and the safety verification framework is not yet fully developed.

The spatiotemporal decoupled framework in the Frenet coordinate system, proposed by Werling et al., significantly improves computational efficiency by optimizing path and velocity in a hierarchical manner[29]. However, it neglects the coupling between space and time, leading to discontinuous trajectory curvature in curved scenarios. In recent years, to overcome this limitation, joint spatiotemporal optimization has gradually become a research hotspot. For example, planning methods based on spatiotemporal corridors generate safe trajectories by constructing spatiotemporal cuboid constraints[30]. However, these methods still suffer from insufficient real-time performance in dynamic environments.

In the field of autonomous driving trajectory planning, although significant progress has been made in previous research, two core challenges in curved road scenarios within structured roads remain to be solved:

Dynamic coupling of spatiotemporal constraints. The traditional Frenet coordinate system can effectively decouple longitudinal and lateral motion planning, but a single reference frame struggles to accurately describe the motion states of traffic participants with different curvature characteristics[31–33].

Co-optimization of planning robustness and efficiency. Existing safety mechanisms often adopt conservative strategies that ensure collision avoidance but frequently result in non-natural driving behaviors such as sudden deceleration[34]. The primary passive safety strategies tend to prioritize collision avoidance at the expense of traffic efficiency, leading to conservative maneuvers that disrupt traffic flow. This safety-first approach may trigger consecutive braking by following vehicles, thereby increasing overall traffic risk and compromising robustness[35–37].

In summary, while previous research has made significant progress in intelligent driving trajectory planning, several limitations still exist, such as limited adaptability and robustness in complex environments, challenges in safety capabilities, and the optimality of joint spatiotemporal methods. The joint spatiotemporal planning algorithm proposed in this paper, which integrates heuristic human-like decision-making thinking, comprehensively considers factors such as road curvature, vehicle posture, and planning cost, effectively addressing the aforementioned issues.

Methods

Spatiotemporal joint planning algorithms

Path planning in the Cartesian coordinate system requires simultaneous optimization of the planar x-y coordinate parameters, which is prone to dimension coupling issues when dealing with complex road geometries. In contrast, the Frenet coordinate system, by decomposing vehicle motion into longitudinal displacement along the road centerline (S-axis) and lateral offset (L-axis), significantly reduces the dimensionality complexity of the planning problem[29]. This decomposition mechanism closely aligns with the cognitive pattern of human drivers regarding lane-relative positions, enabling the system to generate trajectories that are more intuitive for driving[38]. However, the Frenet framework has inherent limitations in multi-agent curved motion scenarios. When the road curvature radius falls below a critical value, a single reference frame can lead to misjudgments of curved steering behavior, mistaking curvature tracking for lane-changing actions[39]. This issue arises from the intrinsic conflict between vehicle steering dynamics and the geometric constraints of the fixed reference frame.

In light of these challenges, this paper proposes a multi-Frenet coordinate system method based on road segmentation, as illustrated in Fig. 1 This method involves concatenating curved roads to "straighten" them. It ensures smooth trajectory planning, widens the spatiotemporal corridor, and reduces the vehicle's envelope radius. Based on this approach, we employ Bezier curves to complete the spatiotemporal trajectory planning. After obtaining the initial trajectory from the decision-making algorithm, each segment of the trajectory is translated using an inverse transformation. The spatiotemporal joint planning algorithm is responsible for transforming a series of inputs from the planning algorithm (including the ego-vehicle's position and state, the ego-vehicle's decision-based coarse trajectory, and the predicted future trajectories of surrounding vehicles) into the Frenet coordinate system and connecting the segments end-to-end. Simultaneously, dynamic and static obstacles are mapped into the grid map, and a three-dimensional feasible convex space in S-L-T is constructed within the grid map. Finally, based on quadratic programming, an optimal trajectory is solved using piecewise Bezier curves within the convex space.

The planning procedure of the spatiotemporal joint planning algorithm is illustrated in Fig. 2, and its technical implementation is divided into four stages:

Road linearization

Drawing on the commonly used road segmentation methods in previous studies on the coupling of road curvature and vehicle body posture, this paper integrates the adaptive discretization concept from dynamic path planning[31,32]. An adaptive curvature-based discretization method is employed to transform the vehicle's state, the coarse-grained trajectory output from the decision-making layer, and the planned trajectories of perceived obstacles into a piecewise Frenet frame. The continuously curved road is divided into N quasi-linear segments, where the number of road segments N is determined by the following formula:

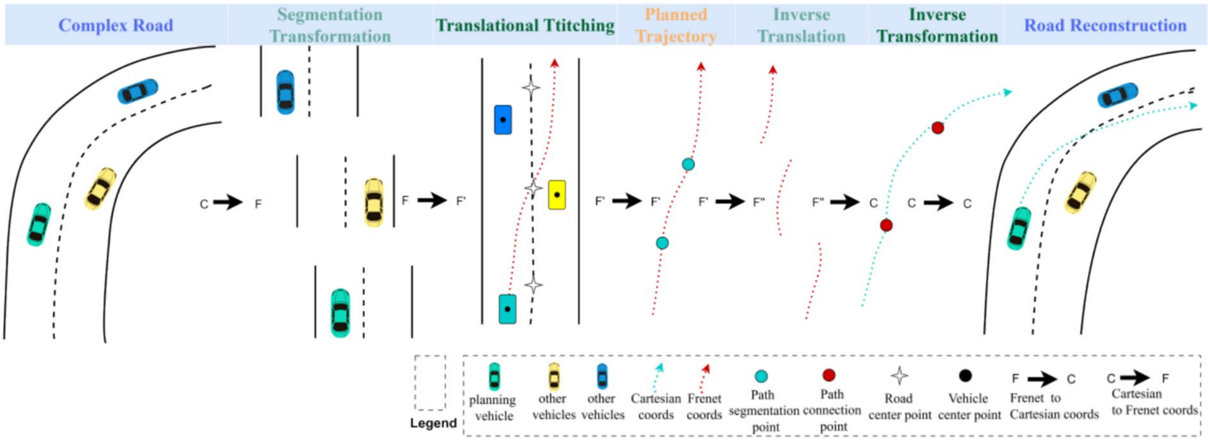


Fig. 1. Schematic diagram of trajectory planning for intelligent vehicles integrating curvature adaptation with spatiotemporal joint optimization.

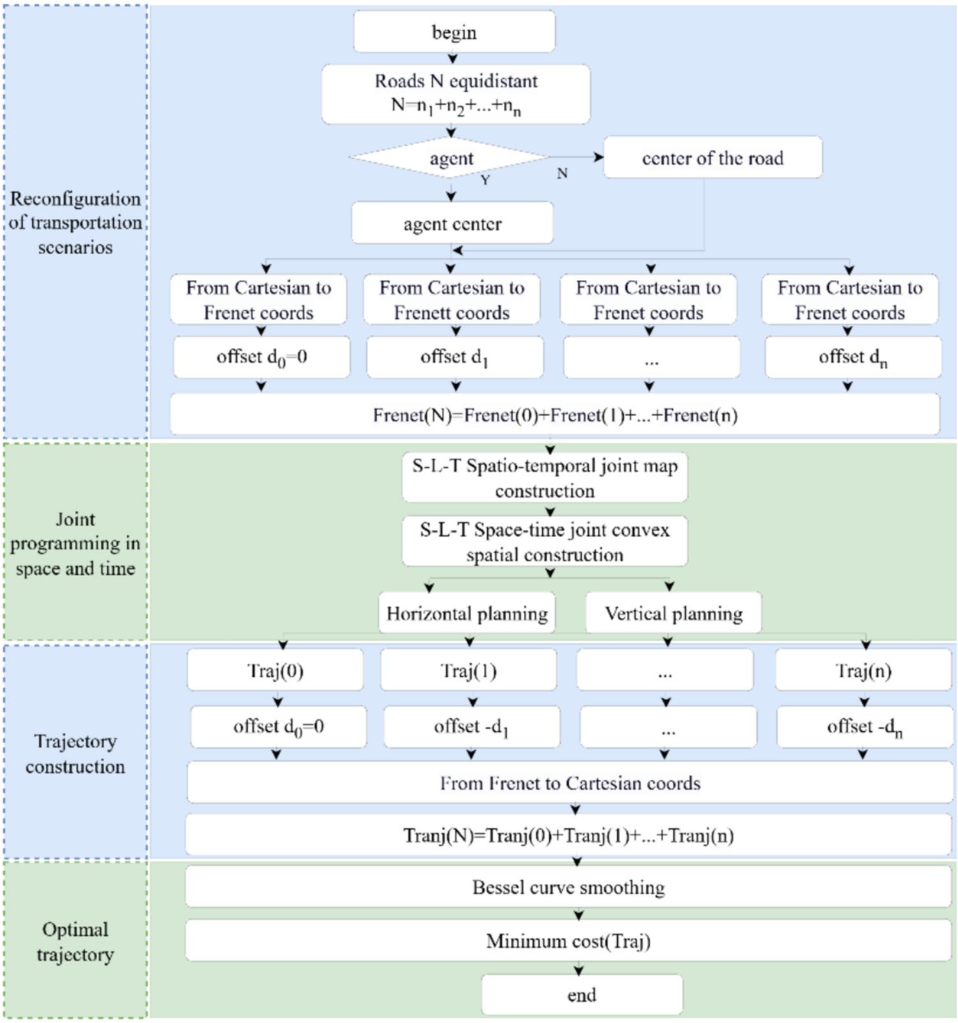

Reconfiguration of transportation scenarios
begin
Roads N equidistant
N=n$_1$+n$_2$+...+n$_n$
agent
Y
N
center of the road
agent center
From Cartesian to Frenet coords
From Cartesian to Frenett coords
From Cartesian to Frenet coords
From Cartesian to Frenet coords
offset d$_0$=0
offset d$_1$
...
offset d$_n$
Frenet(N)=Frenet(0)+Frenet(1)+...+Frenet(n)
Joint programming in space and time
S-L-T Spatio-temporal joint map construction
S-L-T Space-time joint convex spatial construction
Horizontal planning
Vertical planning
Trajectory construction
Traj(0)
Traj(1)
...
Traj(n)
offset d$_0$=0
offset -d$_1$
...
offset -d$_n$
From Frenet to Cartesian coords
Tranj(N)=Tranj(0)+Tranj(1)+...+Tranj(n)
Optimal trajectory
Bessel curve smoothing
Minimum cost(Traj)
end

Fig. 2. The flowchart of the joint spatiotemporal planning algorithm.

$$\begin{gathered} N=\lceil\frac{1}{\Delta\, s_{max}}\int_{s_0}^{s_f}\sqrt{\left(\frac{d\kappa}{ds}\right)^2+\kappa\,^4ds}\rceil \\ \Delta\, s_{max}=\min(\Delta\, s_0,\,\frac{c}{|\kappa\,(s)|+\epsilon}) \end{gathered} \tag{1}$$

Here, Δs_0 denotes the baseline maximum segment length, to balance planning accuracy and computational efficiency, its upper bound is set to the vehicle length. c is a scaling constant (default = 1), and ϵ is a small positive constant to avoid division by zero (default ϵ = 10-6). Consequently, in high-curvature regions, Δs_{max} decreases approximately in inverse proportion to the curvature, enabling finer segmentation when necessary.

$$\overset{max}{\underset{s\in\left[s_i,s_{i+1}\right]}{}}|\kappa(s)|\leq 0.02m^{-1} \tag{2}$$

A local Frenet coordinate system based on the vehicle's center is established for each sub-segment. Through differential geometric transformations, a bidirectional mapping between the global and local coordinates is achieved, thereby forming a piecewise-straightened spatiotemporal planning corridor.

The method of joint spatiotemporal planning

A dynamic feasible region is constructed within the three-dimensional S-L-T (Station-Lateral-Time) space. The spatial occupancy of dynamic and static obstacles is modeled using a convex hull algorithm. For any obstacle, its spatiotemporal occupancy can be modeled as a convex polyhedral constraint in the S-L-T space.

Trajectory construction

The transformed road space is inverse-transformed based on the trajectory generated by the spatiotemporal joint algorithm. The generated trajectory is projected onto the global coordinate system, and then the trajectories of each segment are connected end-to-end (Path connection point). Finally, by applying C^2 continuity constraints at the connections using piecewise Bezier curves, a globally smooth trajectory is synthesized. Let the adjacent trajectory segment $\Gamma_k\Gamma_{k+1}$ be a quartic Bezier curve, with control points $P_k^0, \ldots, P_k^4$ and $P_{k+1}^0, \ldots, P_{k+1}^1$. The R_c continuity must satisfy: the critical curvature radius R meets the vehicle dynamics constraints.

$$R_c = \frac{v^2}{a_{lat}^{max} + \delta_{safe}} \tag{3}$$

Herein, v denotes the vehicle speed, $a_{lat}^{max} = 2.5m/s^2$ represents the maximum allowable lateral acceleration, and δ_{safe} is the safety margin. When the road curvature radius R is smaller than R_c, a multi-coordinate system strategy must be employed.

Quadratic programming solution

Using the control points of the Bezier curve as decision variables, an optimization objective function is established that includes constraints on curvature continuity, jerk, acceleration, velocity, as well as penalty terms for lateral deviation and collision.

Compared with other joint spatiotemporal planning algorithms, the curvature-adaptive joint spatiotemporal planning algorithm operates within a local solution space of the given decision results, thereby avoiding the computational complexity explosion caused by decision diversity. Experiments demonstrate that this method reduces the trajectory envelopment radius compared to traditional single-Frenet-frame approaches. Furthermore, by leveraging parallel computation of subproblems, the planning time is controlled within 0.21s. This divide-and-conquer strategy effectively balances the trade-off between planning accuracy and computational efficiency, offering a novel approach for real-time trajectory planning in complex road scenarios.

The transformation between the Frenet and Cartesian coordinate systems

The Frenet coordinate system

The limitations of the Cartesian coordinate system lie in its inability to directly determine the distance traveled by a vehicle, whether surrounding vehicles have deviated from the lane centerline, or the exact location of the road. In contrast, the Frenet coordinate system is lane-based, where the S-axis represents the distance traveled along the lane, and the L-axis indicates the lateral offset from the lane centerline. Trajectory planning, as a high-dimensional optimization problem with multiple nonlinear constraints, demands significant computational resources. Given the complexity of real-world scenarios and the requirement for real-time processing, it is not always possible to quickly and simply obtain an optimal trajectory. This imposes high demands on the robustness and practicality of planning algorithms. When describing the position of objects using the Cartesian coordinate system, even if the vehicle's location is known, it is challenging to rapidly determine the road's position and whether the vehicle has deviated from the lane center.

In the Frenet coordinate system, however, the road centerline is typically used as the reference line, and the vehicle's trajectory is decoupled into longitudinal and lateral components along and perpendicular to the reference line, respectively. Therefore, compared to the traditional Cartesian coordinate system with x and y axes, the Frenet coordinate system not only provides a more intuitive representation of the road position, especially for curved roads, but also simplifies the two-dimensional planning problem into the superposition of two one-dimensional problems. Additionally, the construction of the Frenet coordinate system avoids the complex transformation of the vehicle's heading direction, which is often required in Cartesian coordinates.

Transformation between Frenet and Cartesian coordinates

The information obtained from the perception module, including road boundaries and obstacles, is represented in the global Cartesian coordinate system. The joint spatiotemporal planning algorithm solves for the optimal trajectory in the Frenet coordinate system. However, the final motion trajectory must be executable and trackable by the control module. Therefore, it is necessary to project the perception information from the global Cartesian coordinate system onto the Frenet coordinate system. Simultaneously, the planned trajectory obtained in the Frenet coordinate system must be transformed back to the global Cartesian coordinate system for execution.

Fig. 3 illustrates the correspondence between the motion trajectory of an intelligent driving vehicle in the Frenet coordinate system and the global coordinate system. Adopting the core idea of the Frenet-Serret framework and based on the geometric transformation between the Frenet coordinate system and the Cartesian coordinates, the following derivation can be obtained[29]. Assuming the position of the intelligent driving vehicle in the global coordinate system is represented as (x, y), the vehicle's position coordinates at the current moment can be expressed as:

$$\overrightarrow{x} = (x,\ y) = \overrightarrow{x}(s,\ l) \tag{4}$$

In the equation, $\overrightarrow{x}$ denotes the position vector of the intelligent driving vehicle in the global coordinate system. s represents the arc length distance from the starting point to the corresponding point on the reference line when the vehicle's position is projected onto it, indicating the longitudinal travel distance of the vehicle along the reference line.   is the lateral offset distance between the vehicle's position and the point corresponding to the longitudinal distance.

The state variables of interest for the vehicle in the Cartesian coordinate system are $[\overrightarrow{x},\ v,\ a,\ \theta_x,\ k_x]$ where v represents the vehicle's velocity in the Cartesian coordinate system, a denotes the vehicle's acceleration in the global coordinate system, θ_x is the vehicle's yaw angle in the Cartesian coordinate system, and k_x represents the curvature. When these state variables are transformed into the Frenet coordinate system, they correspond to $[s, \dot{s}, \ddot{s}, l, l\prime, l'']$.

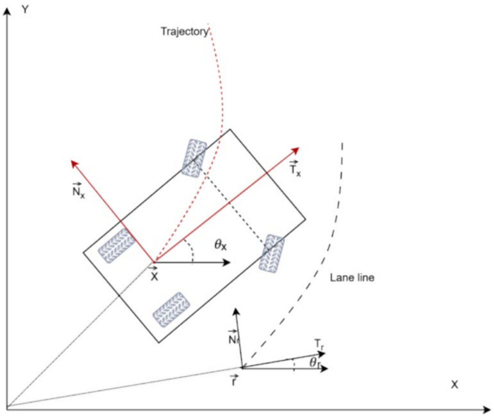


Fig. 3. The illustration of the transformation between the Frenet coordinate system and the global coordinate system.

As shown in Fig. 3, the current position of the vehicle is projected onto the reference line. The reference point at the vehicle center is denoted as $\vec{r} = (x_r,\ y_r)$. The s value of this reference point corresponds to the longitudinal distance of the vehicle in the Frenet coordinate system. $\vec{r}$ represents the position vector of the reference point in the Cartesian coordinate system. The geometric relationship yields:

$$\begin{aligned} \vec{x} &= \vec{r} + l \underset{r}{\vec{N}} \\ l &= (\vec{x} - \vec{r})^T \underset{r}{\vec{N}} \\ &= \left\|\vec{x} - \vec{r}\right\|_2 \cos\left(\theta_{x-r} - \left(\theta_r + \frac{\pi}{2}\right)\right) \\ &= \left\|\vec{x} - \vec{r}\right\|_2 (\sin(\theta_{x-r})\cos(\theta_r) - \cos(\theta_{x-r})\sin(\theta_r)) \end{aligned} \tag{5}$$

In the equation, θ_{x-r} is the direction angle of vector $\vec{x} - \vec{r}$ $\theta_r + \frac{\pi}{2}$; is the direction angle of the normal unit vector $\vec{N}$ of the reference trajectory.

Assuming $\vec{x} = (x_x - y_x)$ and $\vec{r} = (x_x - y_x)$ in the Frenet coordinate system, the vector $\left\|\vec{x} - \vec{r}\right\|_2 = \sqrt{(x_x - x_r)^2 + (y_x - y_r)^2}$ from each coordinate point to its corresponding point on the reference line is either in the same direction or in the opposite direction to the normal vector $\underset{r}{\vec{N}}$ of that reference point. Therefore:

$$\begin{cases} \sin(\theta_{x-r})\, cos\,(\theta_r) - cos\,(\theta_{x-r})\, sin\,(\theta_r) = 1 \theta_r - \theta_{x-r} = \frac{\pi}{2} \\ \sin(\theta_{x-r})\, cos\,(\theta_r) - cos\,(\theta_{x-r})\, sin\,(\theta_r) = -1 \theta_{x-r} - \theta_r = \frac{\pi}{2} \end{cases} \tag{6}$$

$$\frac{\sin(\theta_{x-r})}{\cos(\theta_{x-r})} = \frac{y_x - y_r}{x_x - x_r} \tag{7}$$

Based on the positive or negative relationship of equations (6) and (7), the positive or negative value of the lateral distance   in equation (8) can be determined, which in turn allows us to determine on which side of the reference line the intelligent driving vehicle is at the current moment. Therefore, the formula for calculating the lateral distance   is:

$$l = \text{sign}\left((y_x - y_r)\cos(\theta_r) - (x_x - x_r)\sin(\theta_r)\right)\sqrt{(x_x - x_r)^2 + (y_x - y_r)^2} \tag{8}$$

After determining the longitudinal and lateral distances s and   in the Frenet coordinate system, the longitudinal velocity $\dot{s}$ and longitudinal acceleration $\ddot{s}$ in the Frenet coordinate system can be calculated from the vehicle's speed, yaw angle, and curvature in the global coordinate system. Additionally, the first and second derivatives of the lateral distance with respect to the longitudinal distance, l' and l'' can be computed.

By differentiating the coordinates $(x_x - y_x)^T \underset{r}{\vec{N}}$ with respect to time, we can obtain:

$$i = (\dot{\vec{x}} = \dot{\vec{r}})^T \underset{r}{\vec{N}} + (\vec{x} - \vec{r})^T \underset{r}{\vec{N}} \tag{9}$$

From the geometric relationship of vectors $\vec{x}$ and $\vec{r}$ in Fig. 3, we can derive:

$$\begin{cases} \dot{\vec{x}} = \dfrac{\mathrm{d}\,\vec{x}}{\mathrm{d}t} = \dfrac{\mathrm{d}\left\|\vec{x}\right\|}{\mathrm{d}t} \underset{x}{\vec{T}} = v_x \underset{x}{\vec{T}} \\ \dot{\vec{r}} = \dot{s} l \underset{r}{\vec{T}} \end{cases} \tag{10}$$

In the Frenet coordinate system, the Frenet formulas are required. These formulas are used to describe the motion of a particle along a continuously differentiable curve in the Euclidean space R. They represent the interrelationships among the tangent, normal, and binormal vectors of the curve[14–16]. The specific formulas are as follows:

$$\begin{cases} \dfrac{\mathrm{d}\vec{T}}{\mathrm{d}s} = K\overline{N} \\ \dfrac{\mathrm{d}\vec{T}}{\mathrm{d}s} = -K\overline{T} = \tau\overline{B} \\ \dfrac{\mathrm{d}\vec{B}}{\mathrm{d}s} = -\tau\overline{N} \end{cases} \tag{11}$$

In the formulas, $\overline{T}$ represents the unit tangent vector of the curve at a given point, $\overline{N}$ is the corresponding unit normal vector, and $\overline{B}$ is the binormal vector. k denotes the curvature, and τ represents the torsion. In the context of local trajectory planning for autonomous driving, the road surface can be assumed to be a plane, and thus the torsion τ is generally zero.

Substituting Equations (10) and (11) into Equation (9) yields:

$$i = \left[v_x \vec{T}_x - \dot{s}\,\vec{T}\right]^1 \vec{N}_r - l\,\vec{N}_r^{\,T} ks\,\vec{N}_r \tag{12}$$

Under normal circumstances, the actual trajectory of an autonomous vehicle will always move along the reference line and will not move in the opposite direction of the reference line. Therefore, the vehicle's movement satisfies the following two constraints during its travel:

$$\begin{gathered} -\tfrac{\pi}{2} \le \theta_x - \theta_r \le \tfrac{\pi}{2} \\ 1 < \tfrac{1}{k} \end{gathered} \tag{13}$$

The rate of change of the lateral and longitudinal distances of an autonomous vehicle in the Frenet coordinate system can be derived from the lateral and longitudinal velocities. Combining the constraints in Equation (13), we obtain:

$$l' = \frac{\sqrt{\dot{s}^2(1 - kl) + \dot{l}^2}\sin(\theta_x - \theta_r)}{\dot{s}} = (1 - kl)\tan(\theta_x - \theta_r) \tag{14}$$

By integrating the formulas for the longitudinal and lateral velocities as well as the rate of change of the lateral distance, we can derive the complete transformation relationship between the Frenet coordinate system and the Cartesian coordinate system:

$$\begin{gathered} s = s_r \\ \dot{s} = \tfrac{v_x \cos(\theta_x - \theta_r)}{1 - k_r l} \\ \ddot{s} = \tfrac{a_x \cos(\theta_x - \theta_r) - \dot{s}^2\left[l'\left(k_x \frac{1 - k_r l}{\cos(\theta_x - \theta_r)} - k_r\right) - (k_r' l - k_r l')\right]}{1 - k_r l} \\ l = sign((y_x - y_r)\cos(\theta_r) - (x_x - x_r)sin(\theta_r))\sqrt{(x_x - x_r)^2 + (y_x - y_r)^2} \\ l' = (1 - kl)\tan(\theta_x - \theta_r) \\ l'' = -(k_r' l + k_r l')\tan(\theta_x - \theta_r) + \tfrac{(1 - k_r l)}{\cos^2(\theta_x - \theta_r)}\left(\tfrac{(1 - k_r l)}{\cos(\theta_x - \theta_r)} k_x - k_r\right) \end{gathered} \tag{15}$$

The transformation formula from the Frenet coordinate system to the Cartesian coordinate system is:

$$\begin{gathered} x_x = x_r - lsin(\theta_r) \\ y_x = y_r + l\cos(\theta_r) \\ \theta_x = arctan(\frac{l'}{1-k_r l}) + \theta_r[-\pi, \pi] \\ v_x = \sqrt{\dot{s}(1-k_r l) + (\dot{s}l')^2} \\ a_x = \ddot{s}\frac{1-k_r l}{\cos(\theta_x-\theta_r)} + \frac{\dot{s}^2}{\cos(\theta_x-\theta_r)}[d'(k_x\frac{1-k_r l}{\cos(\theta_x-\theta_r)} - k_r) - (k_r' l + k_r l')] \\ k_x = ((l'' + (k_r' l + k_r l'')\tan(\theta_x - \theta_r))\frac{\cos^2(\theta_x-\theta_r)}{1-k_r l} + k_r)\frac{\cos(\theta_x-\theta_r)}{1-k_r l} \end{gathered} \tag{16}$$

The curvature k_x is calculated in the Frenet coordinate system by taking three consecutive coordinate points on the path and using differential approximation and Taylor expansion to calculate $l\prime$, l'', and $\dot{s}$. These values are then substituted into the following formula to solve for k_x:

$$k_x = \frac{l'}{(1 + l\prime^2)^{\frac{2}{3}}} \tag{17}$$

Similarly, the velocity v_x and acceleration a_x can be calculated by directly using the position and velocity change rates of adjacent path points to approximate them.

Construction of a spatiotemporal joint free feasible convex space

Construction of a three-dimensional Spatiotemporal (S-L-T) joint map

The core idea of constructing a spatiotemporal joint map is to extend the static and dynamic information of vehicles, obstacles, and other elements along the time axis, thereby recording the dynamic changes of environmental information over a period of time, as shown in Fig. 4. The three coordinate axes of the spatiotemporal joint map are the S-axis, L-axis, and T-axis. The S-axis and L-axis are coordinate axes in the Frenet coordinate system, while the T-axis represents the time axis. To construct the spatiotemporal joint map, it is necessary to project the environmental information in the Cartesian coordinate system onto the Frenet coordinate system.

As shown in Equation (18), the map M within the time interval T (starting from t) consists of three parts. At each time layer t, it includes the predicted positions of the surrounding vehicles relative to the ego vehicle, denoted as $\sum_{n=1}^{N_D} P_{t,n}^{D}$, the positions of static obstacles, denoted as $\sum_{n=1}^{N_s} P_{t,n}^{s}$, and the road geometry information, denoted as $\sum_{n=1}^{N_L} R_{l,n}$.

$$\begin{cases} M = \{\sum_{t=t_0}^{T} \sum\limits_{n=1}^{N_D} P_{t,n}^{D}\} \bigcup \{\sum_{t=t_0}^{T} \sum\limits_{n=1}^{N_s} P_{t,n}^{s}\} \bigcup \{\sum_{t=t_0}^{T} \sum\limits_{n=1}^{N_L} P_{t,n}^{L}\} \\ P_{t,n}^{D} = [S_{t,n}, l_{t,n}] \\ P_{t,n}^{s} = [S_{t,n}, l_{t,n}] \\ P_{t,n}^{L} = [L_{w_{t,n}}, L_{id_{t,n}}, n_{s_{t,n}}, n_{l_{t,n}}] \end{cases} \tag{18}$$

In the equation, N_D, N_s, and N_L represent the number of surrounding vehicles, static obstacles, and lanes, respectively. $[S_{t,n}, l_{t,n}]$ denotes the coordinates in the Frenet coordinate system for s and . $[L_{w_{t,n}}, L_{id_{t,n}}]$ represent the width and the number of the lane, respectively. $[n_{s_{t,n}}, n_{l_{t,n}}]$ denotes the Frenet coordinates of the lane centerline.

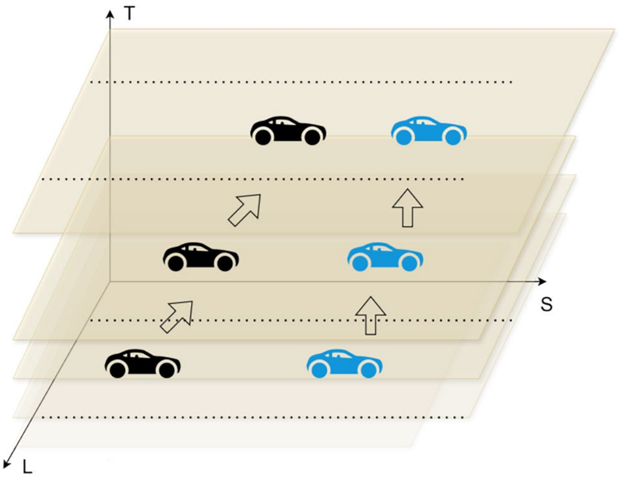


Fig. 4. Schematic diagram of spatiotemporal joint.

Construction of Three-Dimensional Convex Space Based on S-L-T Using a Grid Map

The decision-making algorithm module generated and screened out a feasible coarse-grained trajectory with the minimum cost. Based on the coarse trajectory generated by the decision-making process and the information of static and dynamic obstacles obtained by the perception module, the planning algorithm module performed obstacle filling in the grid map and constructed convex spaces based on the state points on the coarse trajectory obtained from the decision-making process. The planning algorithm module constructed a three-dimensional convex space in the S-L-T domain. The static and dynamic obstacles in the spatiotemporal integrated map are shown in Fig. 5(a-c).

Projection of trajectory into the global coordinate

The decision-making algorithm module generated and screened out a feasible coarse-grained trajectory with the minimum cost. Based on the coarse trajectory generated by the decision-making process and the information of static and dynamic obstacles obtained.

Generation of the coarse trajectory projection of trajectory onto the Cartesian coordinate system

The construction of the three-dimensional convex space in the S-L-T domain mainly involves the following processes: First, the planning algorithm transforms the acquired obstacle information into the Frenet coordinate system using a coordinate transformation method. Subsequently, the transformed obstacle information and the coarse trajectory information obtained from the decision-making process are filled into a grid map of a specified size constructed by the planning algorithm. After completing these tasks, the planning algorithm performs a convex space search based on the coarse-grained trajectory from the decision-making process.

During the search, the algorithm traverses all state points on the coarse trajectory and searches for convex spaces in three directions along the axes. The search continues until a free collision-free space is identified as a convex space, stopping when the boundary of an obstacle is encountered. Fig. 5 illustrates a typical overtaking scenario with lane change, where Fig. 5(a) shows the ego vehicle as a green square, the target lane vehicle as a yellow vehicle, and the preceding vehicle in the current lane as a blue vehicle. The S-T and L-T diagrams for constructing the convex space are shown in Fig. 5(b) and (c), respectively. The red points represent the vehicle state points on the decision-making coarse trajectory, and the constructed convex spaces are depicted as a series of rectangles in Fig. 5(b) and (c). The dashed lines indicate the optimized trajectory.

Projection of trajectory onto the Cartesian coordinate system

To project the generated trajectory back into the real-world traffic coordinate system, the trajectories generated in each Frenet coordinate frame are first subjected to inverse operations based on the original coordinate transformations. Specifically, the trajectories in each Frenet frame are initially translated by an offset of -d. Subsequently, the translated trajectories are transformed from the Frenet coordinate system to the Cartesian coordinate system based on their respective rotation points. Finally, the segments of the transformed coordinates are concatenated end-to-end to form the reference trajectory.

Subsection

Currently, most trajectory generation schemes employ quintic polynomials to fit the lateral and longitudinal trajectories. However, quintic polynomials are not suitable for optimization in spatiotemporal integrated planning algorithms. The primary reason is that the expressive capability of a polynomial segment is limited, making it less appropriate for problems with complex configuration spaces and dynamic constraints. Additionally, in previous studies on polynomial trajectories, constraints were enforced and checked only on a finite set of sampling points. This approach may fail to detect collisions between sampling points, thereby not guaranteeing smoothness

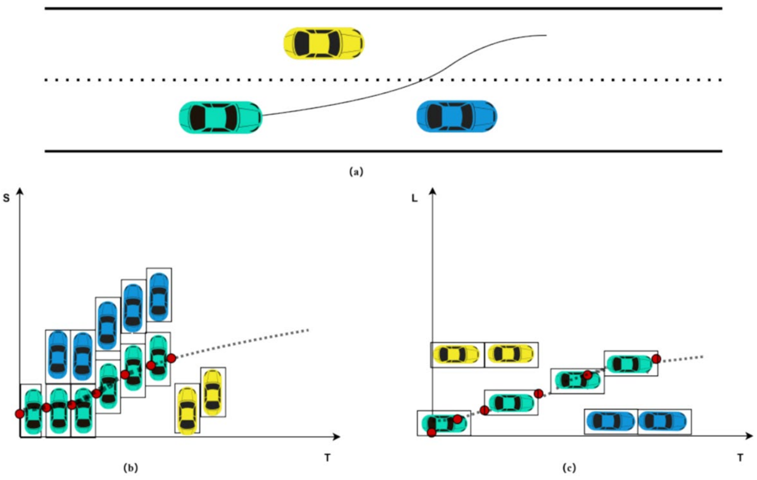


Fig. 5. The planning algorithm module constructed a spatiotemporal integrated convex space.

and optimal solutions. In contrast, Bézier curves, with their convex hull property and convexity-preserving characteristics, can generate fine-grained trajectories and effectively address the aforementioned issues.

Bézier curve

For an m-th order Bézier curve $f(t)$[24], its mathematical expression is:

$$f(t) = P_0 b_m^0(t) + P_1 b_m^1(t) + \cdots + P_m b_m^m(t) = \sum_{i=0}^{m} P_i . b_m^i(t) \tag{19}$$

In the equation, P_i represents the control points of the curve, and $b_m^i(t) = \binom{m}{i} t^i (1-t)^{m-i}$ is the Bézier polynomial.

The set of control points is denoted as P:

$$P = [P_0, P_1, \ldots, P_m] \tag{20}$$

The convex hull property of the Bézier curve is applicable to trajectory optimization problems defined in a convex space. By constraining the curve within the convex space, it can be guaranteed that the generated Bézier curve is collision-free. The convex hull property of the Bézier curve implies that the derivative $\frac{\mathrm{d}f(t)}{\mathrm{d}t}$ of the Bézier curve $f(t)$ remains a Bézier curve. By applying the convex hull property to the derived Bézier curve $\frac{\mathrm{d}f(t)}{\mathrm{d}t}$, the higher-order quantities (velocity, acceleration) of the Bézier curve $f(t)$ can be constrained within the specified limits.

Trajectory optimization based on piecewise bézier curves

To connect the constructed convex spaces, piecewise Bézier curves are employed. To ensure the smoothness and feasibility of the trajectory, an optimization objective function, $cost$ (total cost of the path), is formulated for the piecewise Bézier curve optimization problem. This cost function includes jerk cost, velocity difference cost, time cost, lateral offset cost, and collision cost. Since Bézier curves are defined within a fixed interval, and considering that the acceleration variation of the vehicle during travel should not be excessive to ensure passenger comfort, the cost value of the trajectory is used as the optimization cost function. Specifically, the cost function for the j -th segment of the Bézier curve, $cost_j$, can be expressed as:

$$cost_j = {}_{p}^{min} \sum_{i=1}^{N} \left(\omega_1 \int k\_jerk\mathrm{d}t + \omega_2 k\_t + \omega_3 \int k\_v\mathrm{d}t + \omega_4 \int k\_offset\mathrm{d}t + \omega_5 k\_collision \right) \tag{21}$$

Here, p represents the control point matrix. k_jerk denotes the cost of path smoothness, k_t represents the cost of computation time, k_v denotes the cost of velocity discrepancy, k_offset represents the cost of lateral deviation, and $k\_collision$ denotes the cost of collision.

Jerk cost Denoted by formula $\omega_1 \int k\_jerkdt$, is the integral of the squared jerk (rate of change of acceleration) over the trajectory segment. Jerk reflects the rate of change of acceleration with respect to time. Excessive jerk can cause a jarring sensation during vehicle travel, compromising ride comfort. Therefore, minimizing the jerk cost can further improve the smoothness of the trajectory.

Time cost Denoted by formula $\omega_2 k\_t$, It aims to quantify the temporal cost required for trajectory execution. Minimizing this cost can effectively reduce the travel time along the path and enhance the efficiency of the planning process.

Speed deviation cost Denoted by formula $\omega_3 \int k\_vdt$. The speed deviation cost is utilized to constrain the deviation of vehicle speed from the desired speed, thereby suppressing abrupt acceleration or deceleration. By minimizing the speed deviation cost, the smoothness of the driving process can be ensured, reducing passenger discomfort and avoiding increased energy consumption or handling stability issues caused by severe speed fluctuations.

Lateral offset cost The lateral offset cost, represented by formula $\omega_4 \int k\_offsetdt$, measures the deviation of the vehicle from the target path in the lateral direction. It calculates the squared lateral offset to reflect the lateral distance between the vehicle and the target path. This cost helps ensure that the vehicle stays as close as possible to the ideal trajectory, thereby enhancing the accuracy of the planned path.

Collision cost The collision cost, denoted by formula $\omega_5 k\_collision$, embodies the safety constraints in trajectory planning. It is a function based on the probability of collision risk, evaluating the potential collision hazards between the trajectory segment and surrounding obstacles. By assigning higher weights to segments with higher collision risks, it effectively avoids collisions and ensures the safety of the vehicle during travel.

The weight coefficients ($\omega_1, \omega_2, \omega_3$,$\omega_4$,$\omega_5$) for each cost term are adaptively adjusted through a covariance matrix. This adaptive adjustment mechanism enables the algorithm to dynamically balance the weights of different cost terms according to various driving scenarios and conditions, thereby achieving optimal comprehensive performance in terms of smoothness, efficiency, and safety.

Experiment and analysis

Dataset

The data in this study are based on a parametric sampling strategy for intelligent driving trajectory planning. A multi-dimensional constraint model and a multi-objective optimization function are established to generate safe and efficient data. This method employs a hierarchical planning architecture: in the longitudinal dimension, the speed profile is constructed based on the S-T (space-time) model, with sampling at a resolution of 0.2 s within the time domain [4.5 s, 5.5 s] to generate S-T curves that satisfy the terminal velocity constraints (5.0 – 8.33 m/s). In the lateral dimension, the lateral offset trajectory is generated based on the L-T (lateral-time) model, with discrete sampling at a step length of 1.0 m within the road width of ±8 m, ensuring that the final acceleration is zero. A verification mechanism is established, including dynamic constraints (velocity ≤ 13.89 m/s, acceleration ≤ 8.0 m/s^2, curvature ≤ [curvature limit]), collision detection, and S-L to X-Y coordinate transformation, to filter out physically feasible candidate trajectories. On this basis, a multi-objective cost function is designed to comprehensively evaluate the smoothness of the trajectory (jerk), efficiency (time cost), safety (lateral offset), and velocity tracking accuracy. The candidate trajectories are optimized and ranked through weight allocation.

The experimental environment was established on CentOS, with hardware configurations including a 12th Generation Intel Core i7-12650H processor (base frequency 2.30 GHz), 32 GB DDR5 RAM, and an NVIDIA GeForce RTX 4060 GPU. The algorithm was implemented in Python 3.8, relying on NumPy for matrix operations and Matplotlib for visualization. The experimental design includes the following key elements:

Road model

A circular bidirectional dual-lane environment composed of four curves was constructed. Each lane has a width of 8 m, with a longitudinal length of 100 m and a lateral width of 80 m. The lane boundaries are solid lines, while the center lane line is a dashed line, with lateral boundary constraints of ±8 m.

Obstacle configuration

One obstacle was placed in each curve to increase the difficulty of the bend and simulate scenarios where vehicles intrude into the opposite lane.

Planner parameters

The time sampling interval was set to 0.2 s, with a prediction time domain of [4.5 s, 5.5 s]. The lateral sampling step size was 0.1 m, and the velocity sampling step size was 5 km/h.

Vehicle parameters

A simplified bicycle model was used, with a wheelbase of 2.5 m, vehicle width of 1.8 m, and maximum steering angle of 0.6 rad. Detailed parameters are shown in Table 1.

Trajectory planning experiment in static scenarios

Validation and comparison of models

The integrated space-time model effectively combines the longitudinal and lateral motions of the vehicle. It not only integrates the current state of the vehicle and the surrounding environment but also considers the potential intentions and possible behaviors of other vehicles. Additionally, the model employs curvature-adaptive discretization to straighten the road, effectively addressing planning deviations caused by road curvature and vehicle posture. To validate the effectiveness of the proposed model, a circular simulation scenario with four curved roads was designed, as shown in Fig. 6 and 7. The green lines represent the vehicle's driving trajectory, the yellow lines indicate the planned trajectory set at the current moment, and the red lines denote the real-time optimal trajectory. Fig. 6 illustrates the path planned using the traditional Bézier curve method, while Fig. 7 shows the trajectory planned using the space-time integrated algorithm.

Comparing the experimental results in Fig. 6(a-f) and 7(a-f), it is evident that under the same constraints, the optimal trajectory planned by the traditional Bézier curve successfully bypasses obstacles but maintains a larger clearance from them. In contrast, the trajectory planned by the spatiotemporal coupling algorithm, which employs the temporal-spatial transformation in the Frenet coordinate system, avoids errors caused by vehicle posture and trajectory smoothness. As a result, it provides more precise obstacle avoidance and a smoother operational path. Comparing Fig. 6(a) and 7(a), it can be observed that during the bypass of the first obstacle, the steering angle of the spatiotemporal coupling planning is smaller than that of the traditional Bézier curve planning algorithm, resulting in a smoother driving experience. When comparing the data of other bends and

Parameter name	Symbol	Value	Unit
Scaling factor	K_SIZE	1	-
Front overhang	RF	3.5	m
Rear overhang	RB	1	m
Vehicle width	W	1.8	m
Wheelbase	WB	2.5	m
Maximum steering angle	MAX_STEER	0.6	rad

Table 1. Basic parameters of the vehicle.

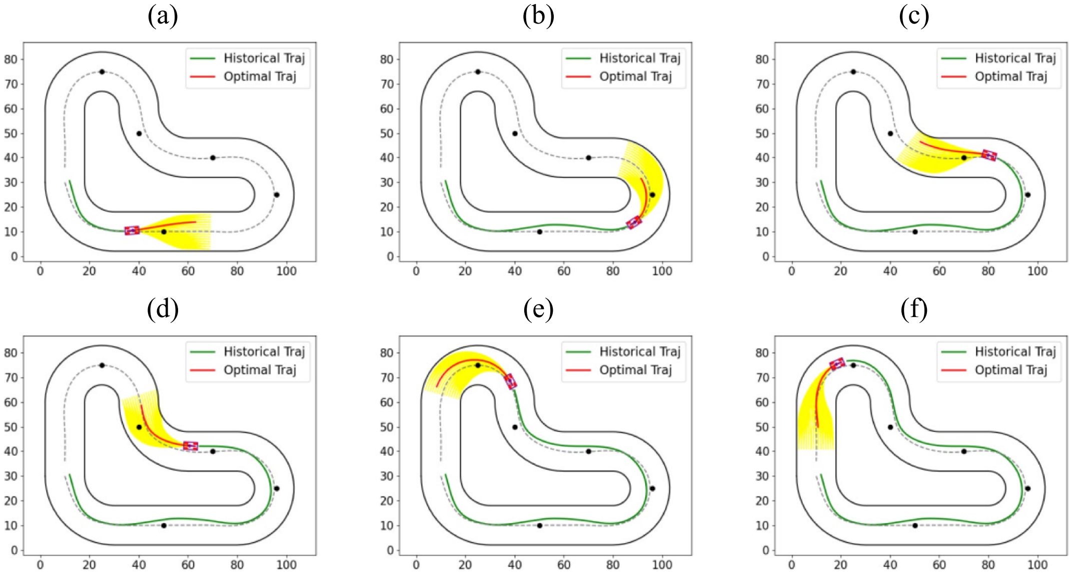


Fig. 6. Trajectory planning based on Bézier curves.

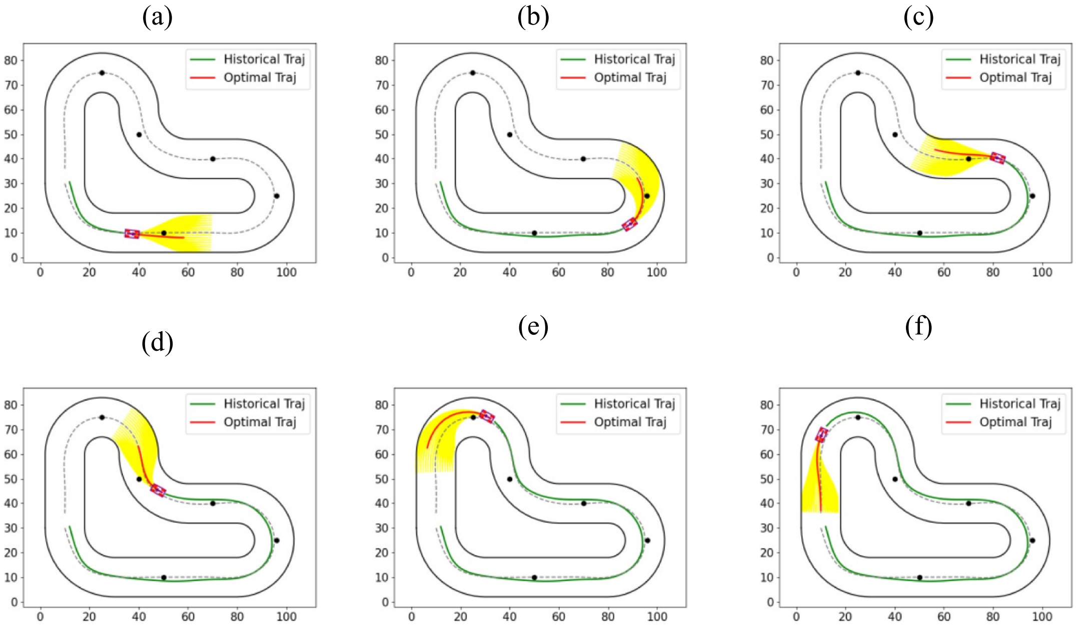


Fig. 7. Trajectory planning based on spatiotemporal integration algorithm.

obstacles, it is apparent that the spatiotemporal coupling algorithm offers a smoother trajectory that is more in line with human driving habits.

Cruise cost analysis

To comprehensively evaluate the superiority of the algorithm, an objective function, cost, is established for single-path optimization. The total cost of the path comprises jerk cost, speed deviation cost, time cost, lateral offset cost, and collision cost. Additionally, cruising cost weights and vehicle operation observation parameters are designed, as shown in Tables 2 and 3.

An intelligent driving trajectory planning method based on a designed detour route and employing a parametric sampling strategy was developed. It achieved secure and efficient data generation through the establishment of a multi-dimensional constraint model and a multi-objective optimization function. The method enabled vehicle simulation and generated a detour cost curve, as shown in Fig. 8.

Experimental results indicate that the algorithm based on spatiotemporal integration has an average cost of 2.229, whereas the trajectory planning method based on traditional Bézier curves has an average cost of

Constraint parameter name	Symbol	Weight	Calculation metric	Affected objective	Priority description
Jerk cost	K_JERK	0.1	Integral of acceleration over time	Comfort	Low
Time cost	K_TIME	0.1	Travel time (seconds)	Efficiency	Medium
Speed deviation cost	K_V_DIFF	0.1	(Target Speed–Actual Speed)^2	Control precision	Medium
Lateral offset cost	K_OFFSET	1.5	Square of lateral offset (m^2)	Trajectory precision	High
Collision cost	K_COLLISION	500	Collision risk probability (0-1)	Safety	Highest

Table 2. Basic parameters of the cost function.

Parameter name	Symbol	Unit
Lateral velocity	L_v	m/s
Lateral acceleration	L_a	m/s^2
Lateral jerk	L_j	m/s^3
Longitudinal velocity	S_v	m/s
Longitudinal acceleration	S_a	m/s^2
Longitudinal jerk	S_j	m/s^3

Table 3. Kinematic parameters of vehicles.

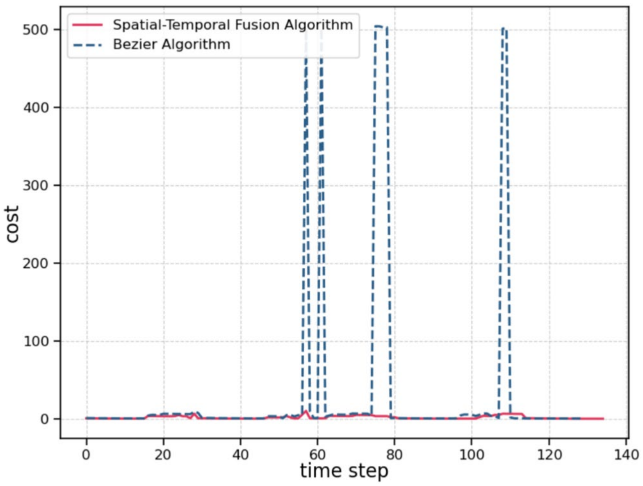


Fig. 8. Comparison of constraint costs.

33.671. It is evident that, despite both methods utilizing Bézier curves, the spatiotemporal integration algorithm successfully planned optimal detour paths without collisions in both single detour and actual driving scenarios. In contrast, although the traditional Bézier curve-based trajectory planning method also successfully navigated around obstacles, it generated collision-prone routes during four specific time intervals in single detour planning. The algorithm considering spatiotemporal integration demonstrated superior performance in terms of jerk cost, velocity deviation cost, time cost, lateral offset cost, and collision cost. It exhibited enhanced robustness in planning, smoothness of the trajectory, and better alignment with human driving habits.

Performance analysis of obstacle-avoidance trajectory planning model

To verify the trajectory planning performance of the spatiotemporal joint model, the Bezier curve planning model proposed in reference[13] was adopted as the comparison model in this study. Under the condition that the basic driving parameters of the intelligent vehicle were identical during the testing process, comparisons were made between the proposed model and the comparison model in terms of longitudinal jerk, longitudinal acceleration, longitudinal velocity, lateral jerk, lateral acceleration, and lateral velocity.

In the longitudinal dimension, compared with Fig. 9(a) and (b), the spatiotemporal joint algorithm exhibits superior performance in terms of both longitudinal jerk and acceleration compared with the reference model, resulting in smoother vehicle motion. As shown in Fig. 9(c) and (f), the speed transition of the spatiotemporal joint algorithm is also more continuous. In the lateral dimension, as depicted in Fig. 9(d) and (e), the lateral acceleration and jerk of the spatiotemporal joint algorithm are slightly higher than those of the reference model, but the differences are minimal. Overall, the spatiotemporal joint algorithm demonstrates higher average speed,

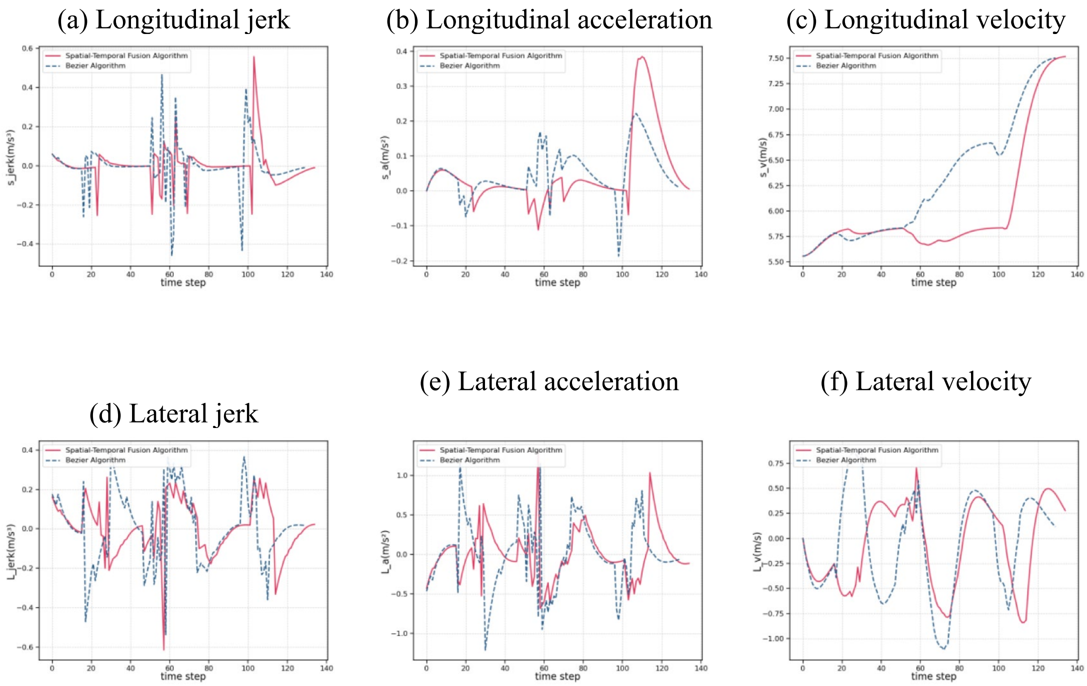


Fig. 9. Comparison of longitudinal and lateral performance.

Algorithm name	Quintic polynomial	Cubic B-spline curve	Cubic Bezier curve	Ours	Fifth-order Bézier curve	Compare with cubic bezier curve	Improved
Optimal planning frequency	133	134	130	135	134	+3.85%	Yes
Execution collision frequency	0	0	0	0	0	0%	-
Planning collision frequency	11	13	8	0	22	-6.15%	Yes
Average longitudinal velocity	6.1548	6.0699	6.2864	6.0350	6.0667	-4.00%	Yes
Average longitudinal acceleration	0.0488	0.04868	0.0499	0.0484	0.0065	-3.01%	Yes
Average longitudinal jerk	0.00077	0.00087	0.00085	0.00077	0.00012	-9.41%	Yes
Average lateral velocity	-0.0537	-0.05691	-0.0588	-0.0595	-0.0496	+1.19%	No
Average lateral acceleration	0.0004	0.0075	0.0020	0.0053	-0.0003	+165.00%	No
Average lateral jerk	-0.0048	-0.0062	0.0032	0.0048	-0.0015	+50.00%	No
Average cost	44.2044	51.7982	33.6714	2.2288	85.4314	-93.38%	Yes
Computation speed (s)	0.0950	0.0940	0.1953	0.2096	0.3249	+7.32%	No

Table 4. Algorithm comparison analysis.

acceleration, and jerk in the longitudinal direction compared with the reference model, indicating enhanced ride comfort during operation. In the lateral motion, the average speed, acceleration, and jerk are all higher than those of the reference model, suggesting more efficient lane-changing maneuvers.

Comparative analysis of spatiotemporal joint model algorithms

The spatiotemporal joint algorithm was compared with quintic polynomial and cubic B-spline algorithms. Based on the comparative experimental data, the spatiotemporal joint algorithm demonstrated significant advantages in several key performance indicators, as detailed in Table 4.

From the comparative data, it can be observed that in terms of trajectory planning density, the spatiotemporal joint algorithm achieved an optimal planning frequency of 135 times, surpassing the quintic polynomial (133 times), B-spline curve (134 times), and Bezier curve (130 times). This indicates its superior global optimization capability. In terms of safety performance, the spatiotemporal joint algorithm successfully achieved zero planning collisions (0 times), compared to the quintic polynomial (11 times), B-spline curve (13 times), and Bezier curve (8 times), demonstrating its robustness in dynamic obstacle avoidance. In longitudinal motion control, the spatiotemporal joint algorithm maintained the lowest average longitudinal acceleration (0.0484 m/s^2) and jerk (0.000766 m/s^3), indicating its ability to generate smoother longitudinal motion trajectories. Although its average longitudinal velocity (6.03502 m/s) is slightly lower than that of the comparative algorithms, it achieves enhanced motion comfort through optimized acceleration control. In lateral dynamics, the spatiotemporal joint algorithm exhibited an average lateral velocity of -0.05945 m/s, which is comparable to the values of the comparative algorithms (-0.0537 to -0.0588 m/s), verifying its comparable lateral motion stability.

Overall, the experimental results demonstrate that the spatiotemporal joint algorithm achieves a better balance in planning efficiency, safety, and motion smoothness. It is particularly suitable for scenarios with high requirements for safety and multi-objective optimization, such as intelligent driving and complex dynamic environments. Despite its relatively lower computational efficiency, the zero planning collisions and minimal comprehensive cost in the planning phase make it a more optimal choice.

Ablation experiments of spatiotemporal joint algorithm

The spatiotemporal joint algorithm employs a curvature-adaptive discretization method to linearize the road. The ego-vehicle state, the coarse-grained trajectory output from the decision-making layer, and the predicted trajectories of perceived obstacles are all uniformly transformed into a piecewise Frenet frame. By dividing the continuously curved road into N quasi-linear segments, the influence of vehicle posture and road curvature on planning is reduced. Details are provided in Table 5.

Based on the data from the comparative experiments, the joint spatiotemporal algorithm demonstrates superior comprehensive performance under the collaborative optimization of road curvature and vehicle attitude. In terms of motion control, the longitudinal velocity of the joint spatiotemporal planning algorithm (6.0350 m/s) is reduced by 4.00% compared with the non-integrated algorithm (6.2864 m/s), while the longitudinal acceleration (0.0484 m/s^2) and jerk (0.00077 m/s^3) are decreased by 3.01% and 9.41%, respectively, thereby effectively enhancing the smoothness of motion. The lateral dynamic indicators remain stable, with the lateral acceleration (0.0053 m/s^2) and jerk (0.0048 m/s^3) being reduced by 7.3%–25.9% compared with the non-joint spatiotemporal algorithm, indicating a significant improvement in the precision of lateral motion control. Regarding planning efficiency, the joint spatiotemporal algorithm achieves a 100% planning success rate across all scenarios with a planning time of 0.2096s, representing a 6.15% increase in success rate compared with the conventional algorithm. Meanwhile, the path cost is significantly reduced from 33.67 to 2.23. The experiments demonstrate that the collaborative optimization strategy integrating road geometric features and vehicle dynamic constraints can significantly enhance the safety and comfort of trajectory planning while ensuring real-time performance.

Trajectory planning experiments in dynamic scenarios

Trajectory planning in dynamic scenarios

To further enhance the validation of the reliability and applicability of our method and to more realistically simulate real-world driving scenarios, experiments involving dynamic obstacle scenarios (lane changing, overtaking, and multi-vehicle environments) are conducted in the following sections.

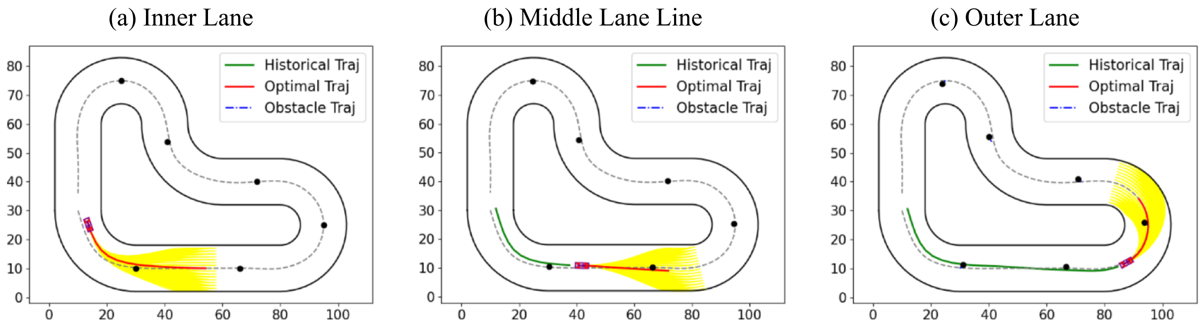
Dynamic scenario - Lane changing We simulate the lane-changing behavior of vehicles, including both overtaking by lane changing and lane changing to avoid obstacles. In the overtaking scenario, the vehicle needs to change lanes at an appropriate time and accelerate to pass the preceding vehicle, thereby verifying the path planning capability and safety of our proposed method during lane changes. In the obstacle-avoidance scenario, the vehicle needs to change lanes promptly upon detecting an obstacle ahead to avoid a collision. While ensuring successful obstacle avoidance, the smoothness and comfort of the vehicle's trajectory are also maintained. The specific simulation results are shown in Fig. 10(a) and (b), which respectively illustrate the lane-changing scenarios in the inner lane, middle lane, and outer lane.

Dynamic scenario – Overtaking We simulate the overtaking behavior of vehicles, including both single overtaking and consecutive overtaking scenarios. Fig. 11 illustrates the entire overtaking process, where Fig. 11(a) and (b) respectively show the planning processes of two consecutive overtaking maneuvers, and Fig. 11(c) shows the completion of the overtaking maneuver. In the single overtaking scenario, the vehicle needs to change lanes at an appropriate time and accelerate to pass the preceding vehicle. In the consecutive overtaking scenario, the vehicle needs to successively overtake multiple slow-moving vehicles ahead. These scenarios are designed to verify the path planning capability of our proposed method during overtaking maneuvers.

Dynamic scenario - Multi-vehicle environment We simulate the scenario where the vehicle encounters multiple dynamic obstacles during its journey. The specific simulation results are shown in Fig. 12, where Fig. 12(a) represents entering a multi-vehicle environment, Fig. 12(b) represents being in the multi-vehicle environment, and Fig. 12(c) represents exiting the multi-vehicle environment. In these scenarios, the motion trajectories and velocities of the dynamic obstacles are randomly generated to simulate the uncertainties in real traffic condi-

Algorithm	Road curvature	Vehicle posture	S_v	S_a	S_J	L_v	L_a	L_J	Time	Cost	Planning success times	Obstacle avoidance success rate
Quartic Bezier curve	/	/	6.2864	0.0499	0.00085	-0.0588	0.0020	0.0032	0.1953	33.67	8/130	100%
	/	√	6.2487	0.0498	0.00087	-0.0650	0.0050	0.0065	0.1962	14.01	3/131	100%
	√	/	6.2461	0.0498	0.00084	-0.0589	0.0045	0.0065	0.2145	18.05	4/131	100%
Ours	√	√	6.0350	0.0484	0.00077	-0.0595	0.0053	0.0048	0.2096	2.23	0/135	100%

Table 5. Ablation studies on the spatiotemporal joint algorithm.

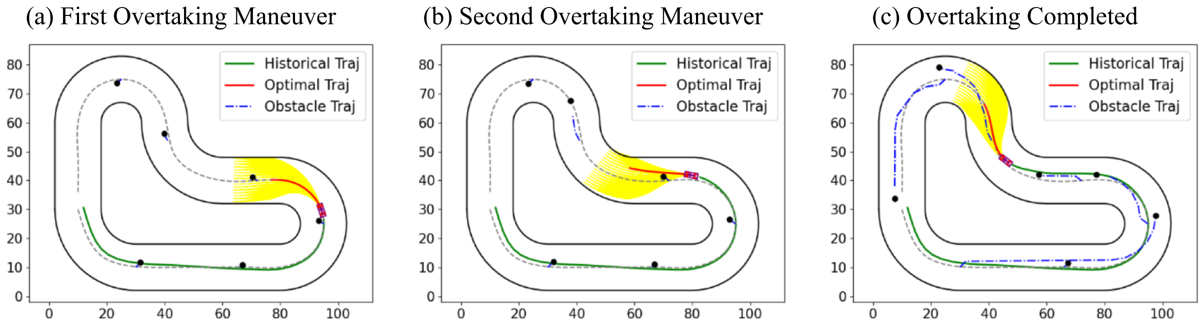


(a) Inner Lane

(b) Middle Lane Line

(c) Outer Lane

Fig. 10. Dynamic scenario - lane changing

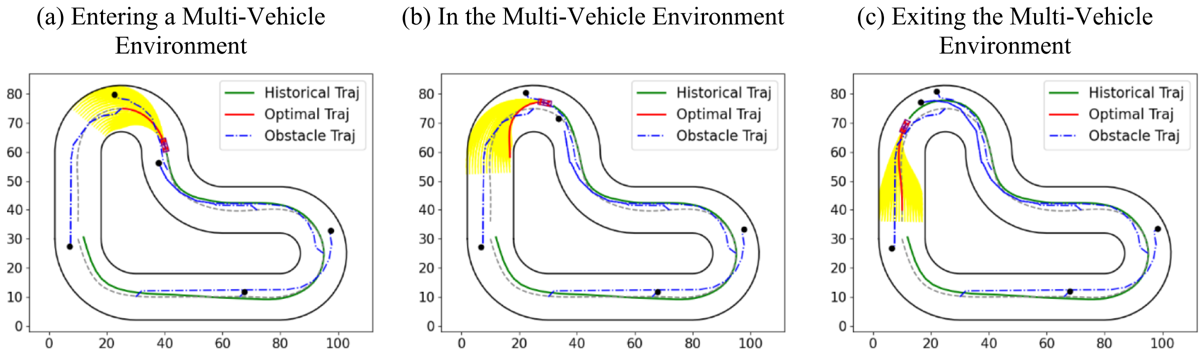


(a) First Overtaking Maneuver

(b) Second Overtaking Maneuver

(c) Overtaking Completed

Fig. 11. Dynamic Scenario – Overtaking.



(a) Entering a Multi-Vehicle Environment

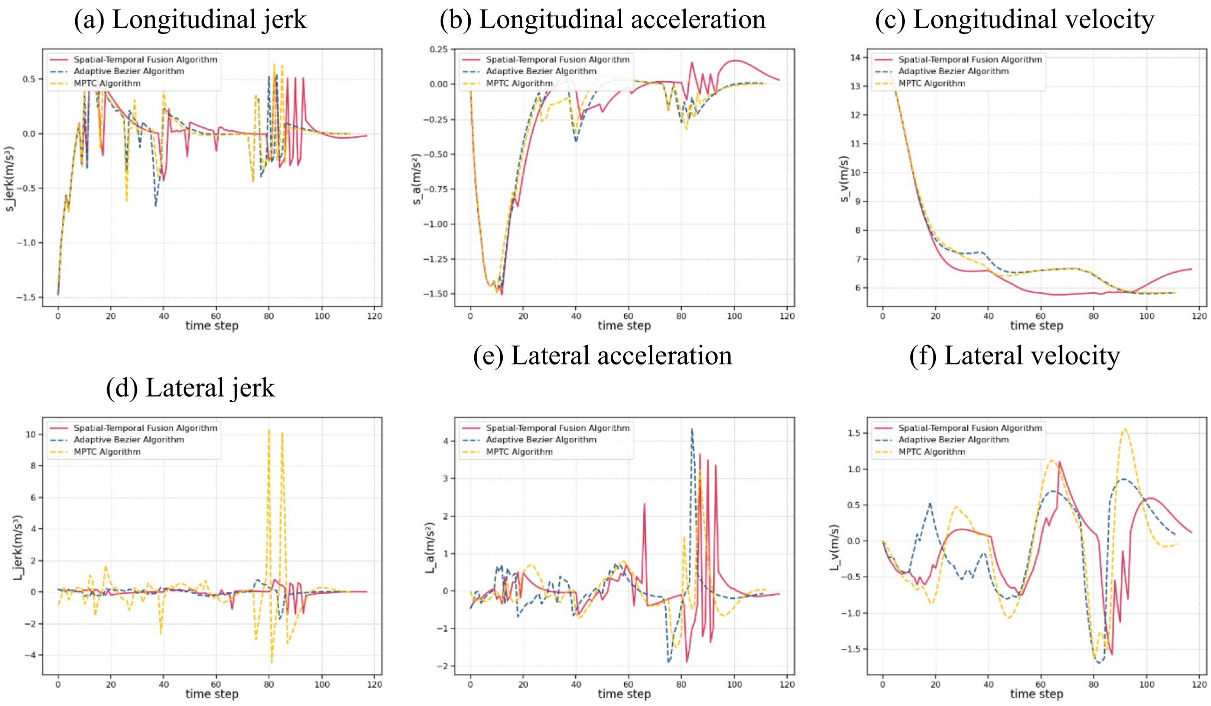
(b) In the Multi-Vehicle Environment

(c) Exiting the Multi-Vehicle Environment

Fig. 12. Dynamic Scenario - Multi-Vehicle Environment.

tions. These experiments are designed to verify the rapid response capability and path re-planning ability of our proposed method in the presence of dynamic obstacles.

This study designed three typical dynamic scenarios: lane-changing scenarios, overtaking scenarios on a straight road, and overtaking scenarios on a curved road. By simulating complex traffic events such as intrusions from the oncoming lane and sudden braking of the preceding vehicle, the results of the experiments, as shown in Figures 10-12, demonstrate that the spatiotemporal joint algorithm performs well in dynamic obstacle avoidance, path smoothness, and planning success rate. In the lane-changing scenario, the target lane is intruded upon by an oncoming vehicle, and the ego vehicle needs to complete the lane change while ensuring safety. The spatiotemporal joint algorithm dynamically constructs a three-dimensional convex space in the S-L-T coordinate system, allowing for rapid trajectory adjustment in response to intrusions and achieving smooth lane changes (Fig. 10(b)). In the overtaking scenario on a straight road, the preceding vehicle is traveling at a low speed, and there is oncoming traffic in the adjacent lane. The spatiotemporal joint algorithm extends the prediction time domain to anticipate the trajectory of the oncoming vehicle and generates a collision-avoidance overtaking path (Fig. 11(b)). In the overtaking scenario on a curved road, which poses dual challenges of road curvature and dynamic obstacles, the spatiotemporal joint algorithm employs road segmentation and adaptive transformation of multiple coordinate systems to generate an optimized trajectory that conforms to the geometric characteristics of the curve (Fig. 12(b)).



(a) Longitudinal jerk

(b) Longitudinal acceleration

(c) Longitudinal velocity

(d) Lateral jerk

(e) Lateral acceleration

(f) Lateral velocity

Fig. 13. Comparison of model planning performance in dynamic environments.

Algorithm name	MPTC	Adaptive bezier	Ours	Improved
Optimal planning frequency	113	112	118	Yes
Average longitudinal velocity	7.2697	7.3024	6.9733	Yes
Average longitudinal acceleration	-0.2376	-0.2399	-0.2045	Yes
Average longitudinal jerk	-0.0032	-0.0032	-0.0021	Yes
Average lateral velocity	-0.0586	-0.0603	-0.0582	Yes
Average lateral acceleration	-0.0010	0.0019	0.0020	No
Average lateral jerk	-0.0005	0.0060	0.0050	No
Average cost	70.1279	74.9516	41.3630	Yes
Computation speed (s)	0.0810	0.2084	0.1881	No

Table 6. Comparison of model planning performance in dynamic environments.

Comparison of model planning performance in dynamic environments

To further validate the trajectory planning performance of the spatiotemporal joint model, this study compared it with several mainstream and representative trajectory planning algorithms, such as the Adaptive Bezier Algorithm[40] and the Model Predictive Trajectory Control (MPTC) Algorithm[41]. The comparison was conducted from multiple aspects, including obstacle avoidance performance and passenger comfort (jerk and lateral acceleration fluctuations). By expanding the dimensions of result analysis and conducting a comprehensive evaluation through quantitative metric comparisons and visual presentations, the results are shown in Fig. 13.

To more clearly illustrate the performance of the algorithms, the basic driving parameters of the intelligent vehicle for both the proposed model and the comparison models were kept identical during the testing process. The longitudinal jerk, longitudinal acceleration, longitudinal velocity, lateral jerk, lateral acceleration, and lateral velocity were separately recorded. The detailed comparisons are shown in Table 6.

Overall, the statistical results clearly demonstrate that in the dynamic scenarios, the proposed spatiotemporal integrated trajectory planning algorithm shows significant improvements in six out of the nine actual indicators evaluated for path planning safety, while the remaining three indicators experience a certain degree of decline. Compared with the MPTC method, the comprehensive cost is reduced by 41.02%, and the planning space is increased by 4.42%. Indicators such as longitudinal smoothness have also been enhanced. Although the lateral acceleration, lateral jerk, and computation time have increased to some extent, the absolute values of these increases are relatively small and have a minimal impact on the actual experience of drivers and passengers. Considering the overall cost comprehensively, the spatiotemporal integrated trajectory planning algorithm exhibits a distinct advantage.

Conclusions

Aiming at the limitations of traditional trajectory planning methods in complex road scenarios, this paper proposes a curvature-adaptive and spatiotemporal integrated trajectory planning algorithm for intelligent vehicles. The algorithm discretizes curved roads into quasi-straight segments using a road segmentation algorithm, constructs a hierarchical planning framework based on the joint optimization of multiple Frenet coordinate systems, and integrates Bezier curve optimization with a multi-objective cost function evaluation. This effectively addresses the key challenge of co-optimizing the dynamic coupling of spatiotemporal constraints and planning robustness.

Experiments show that, in terms of motion control, the spatiotemporal integrated planning algorithm reduces the longitudinal velocity by 4.00% compared to the non-integrated algorithm, with the longitudinal acceleration and jerk decreasing by 3.01% and 9.41%, respectively, thereby significantly improving motion smoothness. Lateral dynamic indicators remain stable, and the precision of lateral motion control is notably enhanced. In terms of planning efficiency, the spatiotemporal integrated algorithm achieves a 100% planning success rate across all scenarios with a planning time of 0.2096 ms. Compared to traditional algorithms, the success rate is increased by 6.15%, and the path cost is significantly reduced from 33.6714 to 2.2288, which is more in line with human driving habits. This algorithm provides a new technological approach for trajectory planning in structured roads for intelligent driving systems. Its divide-and-conquer strategy balances the trade-off between planning accuracy and real-time performance, offering high engineering application value.

Discussion

This framework provides a novel technological approach for trajectory planning in complex road scenarios for intelligent driving systems. Its divide-and-conquer strategy effectively balances the trade-off between planning accuracy and real-time performance, offering significant engineering application value. Future research will focus on further optimizing the computational efficiency of the algorithm to accommodate more complex dynamic environments and scenarios with higher real-time requirements. Additionally, efforts will simultaneously be devoted to enhancing the algorithm's capability to handle dynamic obstacles and its adaptability to complex road conditions such as icy and snowy surfaces, thereby improving its applicability and reliability under intricate traffic scenarios and providing stronger support for the advancement of intelligent driving technologies.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 27 March 2025; Accepted: 7 November 2025
Published online: 21 November 2025

References

1. Ntousakis, I. A., Nikolos, I. K. & Papageorgiou, M. Optimal vehicle trajectory planning in the context of cooperative merging on highways. *Transp. Res. Part C Emerg. Technol.* **71**, 464–488. https://doi.org/10.1016/j.trc.2016.08.007 (2016).
2. Dixit, S. et al. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Ann. Rev. Control* **45**, 76–86. https://doi.org/10.1016/j.arcontrol.2018.02.001 (2018).
3. Yang, D. et al. A dynamic lane-changing trajectory planning model for automated vehicles. *Trans. Res. Part C Emerg. Technol.* **95**, 228–247. https://doi.org/10.1016/j.trc.2018.06.007 (2018).
4. Li, X. et al. Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mech. Sys. Signal Process.* **87**, 118–137. https://doi.org/10.1016/j.ymssp.2015.10.02 (2017).
5. Li, S. et al. Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access* **7**, 132074–132086. https://doi.org/10.1109/ACCESS.2019.2940758 (2019).
6. Madridano, Á., Al-Kaff, A., Martín, D. & de la Escalera, A. Trajectory planning for multi-robot systems: Methods and applications. *Expert Syst. Appl.* **173**, 114660. https://doi.org/10.1016/j.eswa.2021.114660 (2021).
7. You, F. et al. Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system. *Expert Syst. Appl.* **42**(14), 5932–5946. https://doi.org/10.1016/j.eswa.2015.03.022 (2015).
8. Xie, J. et al. Deep reinforcement learning with optimized reward functions for robotic trajectory planning. *IEEE Access* **7**, 105669–105679. https://doi.org/10.1109/ACCESS.2019.2932257 (2019).
9. Zhang, E., Zhang, R. & Masoud, N. Predictive trajectory planning for autonomous vehicles at intersections using reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **149**, 104063. https://doi.org/10.1016/j.trc.2023.104063 (2023).
10. Liu, B. et al. Trajectory planning for autonomous intersection management of connected vehicles. *Simul. Model. Pract. Theor.* **90**, 16–30. https://doi.org/10.1016/j.simpat.2018.10.002 (2019).
11. Xin, L. et al. Enable faster and smoother spatio-temporal trajectory planning for autonomous vehicles in constrained dynamic environment. *Proc. Inst. Mech. Eng. Part. D. J. Automob. Eng.* **235**(4), 1101–1112. https://doi.org/10.1177/0954407020906627 (2021).
12. Li, B. et al. Dynamically integrated spatiotemporal-based trajectory planning and control for autonomous vehicles. *IET Intell. Transp. Syst.* **12**(10), 1271–1282. https://doi.org/10.1049/iet-its.2018.5306 (2018).
13. Guo, J. et al. Spatio-temporal joint optimization-based trajectory planning method for autonomous vehicles in complex urban environments. *Sensor* **24**(14), 4685. https://doi.org/10.3390/s24144685 (2024).
14. Geisslinger, M., Poszler, F. & Lienkamp, M. An ethical trajectory planning algorithm for autonomous vehicles. *Nat. Mach. Intell.* **5**(2), 137–144. https://doi.org/10.1038/s42256-022-00607-z (2023).
15. Li, Q. & Li, X. Trajectory planning for autonomous modular vehicle docking and autonomous vehicle platooning operations. *Transp. Res. Part E Logistics Transp. Rev.* **166**, 102886. https://doi.org/10.1016/j.tre.2022.102886 (2022).
16. Yang, C., Li, Z. & Li, J. Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models. *IEEE Trans. Cybern.* **43**(1), 24–36. https://doi.org/10.1109/TSMCB.2012.2198813 (2012).
17. Aine, S. et al. Multi-heuristic a. *Int. J. Robot. Res.* **35**(1–3), 224–243. https://doi.org/10.1177/0278364915594029 (2016).
18. Burzyński, W. & Stecz, W. Trajectory planning with multiplatform spacetime RRT. *Appl. Intell.* **54**(19), 9524–9541. https://doi.org/10.1007/s10489-024-05650-4 (2024).

19. Li, Q. et al. Smart vehicle path planning based on modified PRM algorithm. *Sensor* **22**(17), 6581. https://doi.org/10.3390/s22176581 (2022).
20. Sun, Q. et al. Human-like obstacle avoidance trajectory planning and tracking model for autonomous vehicles that considers the driver's operation characteristics. *Sensor* **20**(17), 4821. https://doi.org/10.1016/j.heliyon.2023.e14784 (2020).
21. Yang, Y. et al. CSDO: Enhancing efficiency and success in large-scale multi-vehicle trajectory planning. *IEEE Robot. Autom. Lett.* https://doi.org/10.1109/lra.2024.3440091 (2024).
22. Duan, L. et al. Centralized vehicle trajectory planning on general platoon sorting problem with multi-vehicle lane changing. *Transp. Res. Part C Emerg. Technol.* **154**, 104273. https://doi.org/10.1016/j.trc.2023.104273 (2023).
23. Shi, R. & Wang, X. Digitizing traffic rules to guide automated vehicle trajectory planning. *Expert Syst. Appl.* https://doi.org/10.1016/j.eswa.2025.126661 (2025).
24. Wang, L. et al. A review of collaborative trajectory planning for multiple unmanned aerial vehicles. *Processes* **12**(6), 1272. https://doi.org/10.3390/pr12061272 (2024).
25. Guo, Y. et al. Trajectory planning framework for autonomous vehicles based on collision injury prediction for vulnerable road users. *Accid. Anal. Prev.* **203**, 107610. https://doi.org/10.1016/j.aap.2024.107610 (2024).
26. Huang, L. et al. Bidirectional neural network for trajectory planning: An application to medical emergency vehicle. *Neurocomputing* **591**, 127763. https://doi.org/10.1016/j.neucom.2024.127763 (2024).
27. Qin, P. et al. Hierarchical collision-free trajectory planning for autonomous vehicles based on improved artificial potential field method. *Trans. Inst. Meas. Control* **46**(4), 799–812. https://doi.org/10.1177/01423312231186684 (2024).
28. Zhang, H., Yao, J. & Tian, S. Model predictive control for trajectory planning considering constraints on vertical load variation. *Electronic* **13**(8), 1488. https://doi.org/10.3390/electronics13081488 (2024).
29. Werling, M. et al. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation, Anchorage*, 987–993 https://doi.org/10.1109/ROBOT.2010.5509799 (AK, USA, 2010).
30. Toumieh, C. & Lambert, A. Decentralized multi-agent planning using model predictive control and time-aware safe corridors. *IEEE Robot. Autom. Lett.* **7**(4), 11110–11117. https://doi.org/10.1109/LRA.2022.3196777 (2022).
31. Kouhi, H. & Moradi, A. Multiple-vehicle cooperative autonomous parking trajectory planning using connected fifth degree polynomials and genetic algorithm optimization. *IEEE Trans. Intell. Veh.* https://doi.org/10.1109/TIV.2024.3365328 (2024).
32. Sun, J. et al. Nonlinear model predictive control for trajectory-planning and tracking based on tilting technology to achieve vehicle obstacle avoidance. *Veh. Syst. Dyn* **62**(12), 3276–3296. https://doi.org/10.1080/00423114.2024.2326544 (2024).
33. Li, Y., Li, G. & Wang, X. Research on trajectory planning of autonomous vehicles in constrained spaces. *Sensor* **24**(17), 5746. https://doi.org/10.3390/s24175746 (2024).
34. Liu, Y. et al. An integration planning and control method of intelligent vehicles based on the iterative linear quadratic regulator. *J. Franklin Inst.* **361**(1), 265–282. https://doi.org/10.1016/j.jfranklin.2023.11.046 (2024).
35. Shang, Y. et al. Trajectory planning at a signalized road section in a mixed traffic environment considering lane-changing of CAVs and stochasticity of HDVs. *Transp. Res Part C Emerg. Technol.* **158**, 104441. https://doi.org/10.1016/j.trc.2023.104441 (2024).
36. Wu, J. et al. Research on anthropomorphic obstacle avoidance trajectory planning for adaptive driving scenarios based on inverse reinforcement learning theory. *Engineering* **33**, 133–145. https://doi.org/10.1016/j.eng.2023.07.018 (2024).
37. Shafik, A. K. & Rakha, H. A. Integrated back of queue estimation and vehicle trajectory optimization considering uncertainty in traffic signal timings. *IEEE Trans. Intell. Transp. Syst.* https://doi.org/10.1109/tits.2024.3460375 (2024).
38. Du, H. et al. A lane-changing trajectory re-planning method considering conflicting traffic scenarios. *Eng. Appl. Artif Intell* **127**, 107264. https://doi.org/10.1016/j.engappai.2023.107264 (2024).
39. Liu, C. et al. Vehicle trajectory data processing, analytics, and applications: A survey. *ACM Comput. Surv.* https://doi.org/10.1145/3715902 (2025).
40. An, Li. et al. Adaptive bézier curve-based path following control for autonomous driving robots. *Robot. Auton. Syst.* https://doi.org/10.1016/j.robot.2025.104969 (2025).
41. Lai, F. & Huang, C. Seventh-degree polynomial-based single lane change trajectory planning and four-wheel steering model predictive tracking control for intelligent vehicles. *Vehicle* **6**(4), 2624–8921. https://doi.org/10.3390/vehicles6040109 (2024).

Author contributions

Author Contributions Yatao Zhao: Conceptualization, Methodology, Software, Writing - Original Draft, Supervision, Project Administration. Jiangfeng Wang: Review & Editing. Peikun Li: Resources, Investigation. Hao Wang: Review & Editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.