A Two-Stage Spatiotemporal Trajectory Optimization Framework for Autonomous Lane Changing with Dynamic Risk Fields

Zhihao Lin[1], Zhen Tian[1], Xianxian Zhao[2,§], Hanyang Zhuang[3], *Member, IEEE*, Ming Yang[4], *Member, IEEE*, and Jianglin Lan[1]

***Abstract*—Autonomous vehicles (AVs) face challenges in making accurate decisions and planning optimal trajectories in complex environments. Current methods often overlook future dynamics or are too computationally demanding for real-world use, with many studies limited to simple scenarios. To address these limitations, this paper introduces a two-stage trajectory planning framework that considers safety, efficiency, and ride comfort simultaneously in dynamic environments. The key contribution is a novel Dynamic Risk Field (DRF) that provides a unified representation of static and dynamic obstacles, enabling comprehensive risk assessment without treating them separately. The first stage generates spatially feasible trajectories using quintic polynomials in the Frenet frame, guided by the DRF to assess risks from obstacles. The second stage optimizes these trajectories temporally with Space-Time (ST) graphs and Sequential Quadratic Programming (SQP), ensuring collision avoidance and smooth dynamics. Compared to existing methods, our framework achieves efficient real-time planning through the two-stage decomposition while maintaining trajectory quality. Simulations across highway lane changes, overtaking, and intersection scenarios demonstrate that the framework produces safe, efficient, and comfortable trajectories, outperforming benchmark algorithms in terms of acceleration and jerk metrics.**

***Index Terms*—Autonomous driving, dynamic risk field, lane changing, sequential quadratic programming, space-time graph.**

I. INTRODUCTION

AUTONOMOUS vehicles (AVs) are increasingly popular in transportation. Unlike human-driven vehicles (HDVs), AVs can significantly reduce incidents caused by human errors, such as poor driving habits and delayed reactions [1]. Equipped with advanced computing devices, AVs enhance decision-making accuracy [2]–[4]. Additionally, they can increase road capacity and efficiency. Thus, developing safe and efficient planning for AVs is important for the future of intelligent transportation [5].

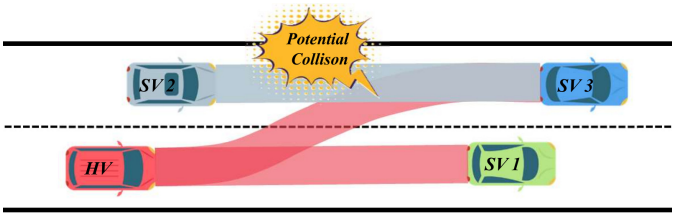


Fig. 1: How can the autonomous host vehicle (HV) interact with surrounding vehicles (SVs) to make accurate decisions and plan collision-free trajectories during lane changing?

Interactive driving between AVs and HDVs is crucial for safe and efficient planning in complex scenarios, such as lane changes with multiple surrounding vehicles (SVs), as illustrated in Fig. 1. During such maneuvers, the autonomous host vehicle (HV) faces collision risks with several SVs. For instance, following SV 1 may cause the HV to miss the optimal lane-change point, while a slow lane change could lead to collisions with SV 2 [6]. Even avoiding collisions can compromise efficiency and increase travel time. This highlights the challenges of interactive driving in complex scenarios like lane changes, where AVs must seamlessly integrate with traffic flow [7]–[10]. Research has shown that lane changing requires the AVs to achieve precise coordination with HDVs and flexible decision-making to adapt to the speed and position of surrounding vehicles. If no suitable lane-change opportunity arises, the AV should follow the vehicle ahead, adjusting its acceleration accordingly [11]. However, treating lane changing and car-following as separate processes often leads to inconsistent behaviour [12], underlining the need for a continuous, human-like driving approach better suited to real-world conditions. In interactive scenarios, AVs must consider both dynamic HDVs and static obstacles while selecting an optimal trajectory that avoids collisions [13]–[15]. Therefore, balancing both current and future situation-based planning, without relying solely on car-following maneuvers, is key to effective interactive driving [16].

Current situation-based planning is essential for ensuring AVs reach target points safely with stability. Several methods have been proposed for trajectory generation in such contexts [17]. The Rapidly-exploring Random Tree (RRT) approach

This work was supported in part by the China Scholarship Council Ph.D. Scholarship for 2023-2027 (No.202206170011), in part by the Leverhulme Trust Early Career Fellowship (ECF-2021-517), in part by the UK Royal Society International Exchanges Cost Share Programme (IEC\NSFC\223228), and in part by the SEAI (Sustainable Energy Authority of Ireland) under RD&D Award 22/RDD/776.

[1]Zhihao Lin, Zhen Tian, and Jianglin Lan are with the James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, United Kingdom

[2]Xianxian Zhao is with the School of Electrical and Electronic Engineering, University College Dublin, Belfield, D04 V1W8 Dublin, Ireland

[3]Hanyang Zhuang is with the Global College, Shanghai Jiao Tong University, Shanghai, 200240, China

[4]Ming Yang is with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, 200240, China

[§]Corresponding author. Xianxian Zhao(e-mail: xianxian.zhao@ucd.ie)

[18] effectively solves motion planning problems by creating a space-filling tree to navigate between states while avoiding obstacles. However, RRT suffers from uncertain convergence rates, often leading to suboptimal solutions [19]. Another approach is the Voronoi diagram-based method [20], [21], valued for its simplicity, versatility, and efficiency. Yet, its performance depends heavily on cell distribution, and it becomes less effective in areas with sparse cells. The Artificial Potential Field (APF) method [22], which uses attractive and repulsive forces to guide vehicles, is also used in environments with multiple obstacles. However, APF often results in unstable trajectories due to the interaction of these forces [23]. In summary, existing current situation-based methods (e.g., RRT, Voronoi diagrams, and APF) focus primarily on the current state of the environment but struggle to anticipate and adapt to future changes, especially with dynamic obstacles [24].

Future situation-based planning enhances the robustness of AV trajectory generation by anticipating potential changes in the obstacle landscape [25], [26]. Nonlinear model predictive control (NMPC) [27]–[29] is a representative approach that predicts future states and repeatedly solves constrained optimization problems, but it is often computationally demanding and sensitive to uncertainty [30]. Monte Carlo Tree Search (MCTS) [31] explores future outcomes through simulation, offering strong performance in large and uncertain state spaces; however, its tight coupling of decision-making and planning can make error correction difficult once suboptimal branches are expanded. Space-time planning [26], [32] jointly optimizes spatial and temporal profiles to handle dynamic obstacles, but remains computationally expensive and has mostly been validated in simplified highway settings. Overall, while future situation-based methods provide improved foresight over reactive planners, they still face challenges in computational efficiency and deployment in complex real-world environments.

Recent studies have explored spatiotemporal heuristic search methods, such as the spatiotemporal-restricted A* algorithm for lane-free traffic at intersections [33], as well as deep reinforcement learning (DRL) approaches for highway autonomous vehicle control [34], [35]. However, these methods are often tailored to specific scenarios or require extensive training data. This motivates the development of optimization-based planning frameworks that can efficiently handle diverse and dynamic driving environments.

To address the challenges, this paper introduces a two-stage optimization approach. The upper stage focuses on initial planning using spatial-inference features, while the lower stage refines planning based on the selected candidate paths and time-inference features. The main contributions are as follows:

1) A novel Dynamic Risk Field (DRF) is proposed to unify the representation of static and dynamic obstacles, enabling risk assessment in complex driving environments.
2) A two-stage trajectory planning framework is developed, integrating spatial trajectory generation via quintic polynomials in the Frenet frame with temporal optimization on Space-Time (ST) graphs under a multi-objective receding-horizon formulation.
3) The resulting constrained nonlinear trajectory optimization problem is efficiently solved using Sequential Quadratic Programming (SQP), allowing robust handling of nonlinearities and safety constraints in dynamic autonomous driving scenarios.

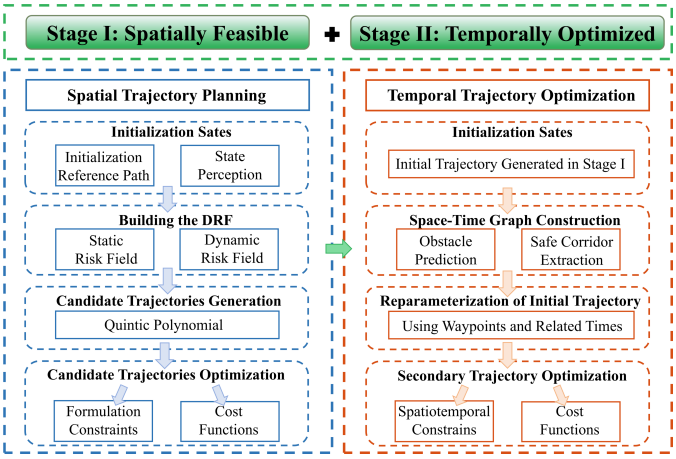


Fig. 2: The workflow of our proposed system.

II. Problem Description and System Structure

Let $\mathbf{x}(t) = [s(t), d(t), \dot{s}(t), \dot{d}(t), \ddot{s}(t), \ddot{d}(t)]^\top$ be the vehicle state at time t, where $s(t)$ and $d(t)$ represent the longitudinal and lateral positions in the Frenet frame, respectively. The vehicle dynamics are governed by

$$\dot{x} = v\cos(\theta),\ \dot{y} = v\sin(\theta),\ \dot{\theta} = v\tan(\delta)/L,\ \dot{v} = a \quad (1)$$

where (x, y) is the global position, θ is the heading angle, v is the velocity, δ is the steering angle, L is the wheelbase, and a is the acceleration.

The AV trajectory planning in dynamic environments can be formulated as a constrained optimization problem. Given a reference path Γ, sets of static and dynamic obstacles $\mathcal{O}_s$ and $\mathcal{O}_d$, a planning horizon T, and an initial vehicle state $\mathbf{x}_0$, the aim is to find an optimal trajectory $\boldsymbol{\gamma}^*$ that minimizes a cost function while satisfying constraints. The cost function $J(\boldsymbol{\gamma})$ is defined as a weighted sum of multiple objectives as follows:

$$\begin{aligned} J(\boldsymbol{\gamma}) = {} & w_1 J_{\text{time}}(\boldsymbol{\gamma}) + w_2 J_{\text{smoothness}}(\boldsymbol{\gamma}) + w_3 J_{\text{safety}}(\boldsymbol{\gamma}) \\ & + w_4 J_{\text{comfort}}(\boldsymbol{\gamma}) + w_5 J_{\text{efficiency}}(\boldsymbol{\gamma}) \end{aligned} \quad (2)$$

where $J_{\text{time}}, J_{\text{smoothness}}, J_{\text{safety}}, J_{\text{comfort}}$ and $J_{\text{efficiency}}$ represent the various costs to be balanced, with corresponding weights $w_i, i \in [1, 5]$, as detailed in Section IV-C. To tackle the complexity of trajectory planning, we propose a two-stage spatiotemporal planning framework (outlined in Fig. 2):

1) **Stage I: Candidate Trajectory Generation.** We generate a set of candidate trajectories in the Frenet frame using quintic polynomials of time, which define both the geometric path and an initial time-parameterized motion profile. This stage produces kinematically feasible trajectories that satisfy dynamic constraints and avoid high-risk regions identified by the DRF (Section III).
2) **Stage II: Spatiotemporal Refinement.** The candidate trajectory from Stage I is further refined in the space-time domain using an ST graph. This stage re-optimizes

the velocity profile and time allocation along the selected path via SQP, explicitly resolving time-dependent interactions with dynamic obstacles while satisfying speed and acceleration constraints. The details are in Section IV.

III. DRF-based Candidate Trajectory Generation

This section introduces a novel path planning algorithm (Algorithm 1) that integrates a dynamic Frenet frame, comprehensive risk assessment, and trajectory optimization. The algorithm effectively tackles real-time path planning challenges involving both static and dynamic obstacles while ensuring vehicle kinematic constraints and passenger comfort. We define a reference path Γ as the baseline for optimization and risk assessment. By transforming the vehicle motion into a local coordinate system relative to Γ, we simplify the planning problem to focus on longitudinal and lateral deviations. The details of the algorithm are provided below.

A. Reference Path Γ Definition

We define Γ using discrete points $(x_i, y_i)_{i=1}^{N}$ in the global Cartesian coordinate system. These points are interpolated using a cubic spline to create a continuous, twice-differentiable reference path $\mathbf{r}(s) = [x_r(s), y_r(s)]^\top$, where s is the arc length along the path. The curvature $\kappa(s)$ of Γ is given by

$$\kappa(s) = \frac{\dot{x}_r(s)\ddot{y}_r(s) - \dot{y}_r(s)\ddot{x}_r(s)}{(\dot{x}_r(s)^2 + \dot{y}_r(s)^2)^{3/2}} \tag{3}$$

where $x_r(s)$ and $y_r(s)$ are the x and y coordinates of Γ, and $\dot{x}_r(s), \ddot{x}_r(s)$ and $\dot{y}_r(s), \ddot{y}_r(s)$ are the first-order and second-order derivatives of Γ with respect to s.

The transformation between Frenet coordinates (s, d) and Cartesian coordinates (x, y) is defined as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_r(s) \\ y_r(s) \end{bmatrix} + d \begin{bmatrix} -\sin\theta(s) \\ \cos\theta(s) \end{bmatrix} \tag{4}$$

where d represents the lateral offset from Γ and $\theta(s) = \arctan 2(\dot{y}_r(s), \dot{x}_r(s))$ is the heading angle of Γ at s.

B. Dynamic Risk Field (DRF) Construction

To ensure safe navigation in dynamic environments, we propose a DRF to quantify the risk associated with the vehicle's current and future positions. Conventional APF methods treat static and dynamic obstacles separately and often suffer from local minima [22]. Our DRF provides a unified representation for both obstacle types. Moreover, the proposed DRF incorporates velocity-dependent scaling and time-varying predictions, enabling more accurate risk assessment in multi-obstacle scenarios. We define the total risk field $R(s, d, t)$ as

$$R(s, d, t) = R_s(s, d) + R_d(s, d, t) \tag{5}$$

with static and dynamic risk fields $R_s(s, d)$ and $R_d(s, d, t)$.

The static risk field $R_s(s, d)$ accounts for road boundaries and static obstacles and it is described as

$$R_s(s, d) = \sum_{i=1}^{N_s} A_i \exp\left(-\frac{(s - s_i)^2}{2\sigma_{s,i}^2} - \frac{(d - d_i)^2}{2\sigma_{d,i}^2}\right) \tag{6}$$

Algorithm 1 Receding Horizon Trajectory Planning

```
Initialize vehicle state (s_0, d_0, ṡ_0, ḋ_0, s̈_0, d̈_0), and Γ.
Set receding horizon parameters: ΔT, T_max;
tolerance for constraint violation ξ_c; minimum iterations k_min; threshold for cost function value change ξ_j.
while not at goal do
    Update DRF R(s, d, t) in (7) for all obstacles.
    Sample end states (s_e, d_e, T_e) within ranges:
s_e ∈ [s_min, s_max], d_e ∈ [d_min, d_max], T_e ∈ [T_min, T_max].
    for each sampled end state do
        Generate candidate trajectory τ using (9).
        if τ satisfies constraints in (11) then
            Add τ to the feasible trajectories set T.
        end if
    end for
    Initialize τ_0 as best trajectory from T.
    Initialize Lagrange multipliers λ_0.
    Initialize Hessian approximation H_0.
    for k = 0 to max_iterations do
        Solve QP subproblem (13) for d_k.
        Perform line search to determine step size α_k.
        Update H_{k+1} from (14).
        Update τ_{k+1} from (15).
        Update λ_{k+1} from (16).
        if ( then|J(τ_{k+1}) − J(τ_k)| < ξ_j) AND (max constraint violation < ξ_c) AND (k > k_min)
            Return optimal trajectory τ* and break.
        end if
    end for
    Execute initial portion of τ* for duration ΔT.
    Update vehicle state and obstacle information.
    Shift optimization horizon.
end while
```

where (s_i, d_i) denotes the position of the i-th static obstacle or boundary point, A_i represents the maximum risk value, and $\sigma_{s,i}$ and $\sigma_{d,i}$ are scaling factors that control the spread of the risk field in the s and d directions, respectively.

The DRF $R_d(s, d, t)$ incorporates the predicted motion of dynamic obstacles and is expressed as

$$R_d(s, d, t) = \sum_{j=1}^{N_d} \frac{B_j \exp\left(-\frac{(s - s_j(t))^2 + (d - d_j(t))^2}{2\sigma_j^2(v_{rel,j})}\right)}{1 + \exp(-k_j(v_{rel,j})(\Delta s_j(t) - \alpha L_j))} \tag{7}$$

where $(s_j(t), d_j(t))$ denotes the predicted position of the j-th dynamic obstacle at time t, B_j is the maximum risk magnitude, $v_{rel,j}$ is the relative velocity, $\Delta s_j(t)$ is the longitudinal distance, L_j is the obstacle length, and α is a safety factor. Although the HV is approximated as a point mass [36], sufficient clearance is ensured through the spatial spread $\sigma_j(\cdot)$ and the safety margin α. The velocity-dependent spread $\sigma_j(v_{rel,j})$ and sensitivity $k_j(v_{rel,j})$ are defined as follows:

$$\begin{aligned} \sigma_j(v_{rel,j}) &= \sigma_0 + k_v|v_{rel,j}|, \\ k_j(v_{rel,j}) &= k_0(1 + k_r|v_{rel,j}|), \end{aligned} \tag{8}$$

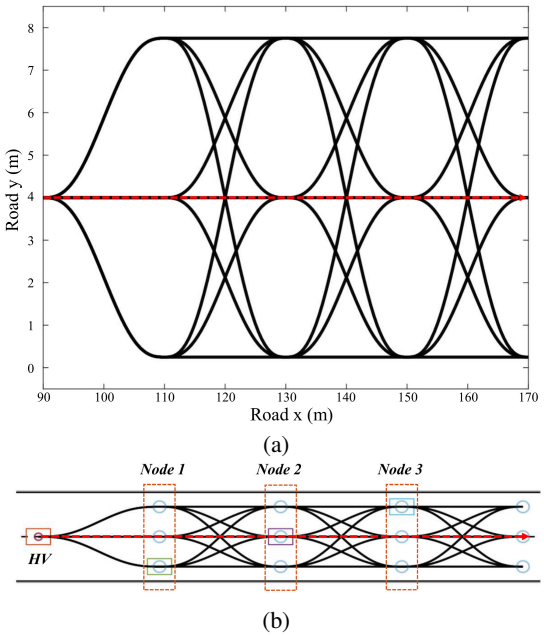


Fig. 3: Candidate trajectories generated using quintic polynomials in the Frenet coordinate system. (a) Trajectories for multiple lane changes. (b) Sampling end states at different nodes to create diverse candidate trajectories.

where σ_0 is the baseline risk spread determined by the obstacle geometry, and k_v is a velocity scaling coefficient. k_0 is the baseline sensitivity and k_r is a relative velocity scaling factor.

C. Candidate Trajectories Generation

Candidate trajectories are generated considering the vehicle's dynamic constraints and the DRF-based risk assessment (Step 5 in Algorithm 1). Incorporating the risk assessment ensures the algorithm select the optimal path that balances safety, efficiency, and comfort. We denote the complete vehicle state trajectory as $\boldsymbol{\tau}$, which includes the position trajectory $\mathbf{p}(t) = [s(t), d(t)]^\top$ and its derivatives up to third order, where $s(t)$ and $d(t)$ are the longitudinal and lateral positions along Γ, respectively. To ensure smooth transitions and continuity up to the third-order derivatives, $s(t)$ and $d(t)$ are generated using quintic polynomials, which can exactly satisfy six boundary conditions (position, velocity, and acceleration at both endpoints) [36], as follows:

$$s(t) = \sum_{i=0}^{5} a_i t^i, \quad d(t) = \sum_{i=0}^{5} b_i t^i \tag{9}$$

where a_i and b_i are the coefficients to be determined. These coefficients are solved by imposing the following longitudinal boundary condition $\mathbf{s}_{\text{boundary}}$ and the lateral boundary condition $\mathbf{d}_{\text{boundary}}$ at the initial time t_0 and final time t_f:

$$[s(t_0), \dot{s}(t_0), \ddot{s}(t_0), s(t_f), \dot{s}(t_f), \ddot{s}(t_f)] = \mathbf{s}_{\text{boundary}},$$
$$[d(t_0), \dot{d}(t_0), \ddot{d}(t_0), d(t_f), \dot{d}(t_f), \ddot{d}(t_f)] = \mathbf{d}_{\text{boundary}}.$$

The coefficients $a_i, i \in [0, 5]$, are then solved from

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \mathbf{s}_{\text{boundary}}. \tag{10}$$

A similar equation using $\mathbf{d}_{\text{boundary}}$ is used to determine the coefficients $b_i, i \in [0, 5]$.

After determining the coefficients, we generate different trajectories by sampling end states (s_e, d_e, T_e) within predefined ranges (Step 6 in Algorithm 1): $s_e \in [s_{\min}, s_{\max}]$, $d_e \in [d_{\min}, d_{\max}]$, $T_e \in [T_{\min}, T_{\max}]$. The longitudinal range $[s_{\min}, s_{\max}]$ is determined by the vehicle's current velocity and maximum acceleration capability, ensuring reachable positions within the planning horizon. The lateral range $[d_{\min}, d_{\max}]$ corresponds to the available lane positions (e.g., center lines of adjacent lanes). The temporal range $[T_{\min}, T_{\max}]$ is set based on typical lane changing durations observed in naturalistic driving, with $T_{\min}$ ensuring kinematic feasibility and $T_{\max}$ limiting computational cost. This process is illustrated in Fig. 3, where each node represents a sampled end state on a different lane. The sampling intervals balance efficiency and trajectory diversity. We then select the optimal candidate trajectory that meets spatial constraints and driving comfort requirements by formulating the spatial trajectory generation as an optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\tau}} J(\boldsymbol{\tau}) \\ \text{s.t.} \quad &\boldsymbol{\tau} = [\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}, \dddot{\mathbf{p}}], \ \mathbf{p}(t) = [s(t), d(t)]^\top, \\ &s(t) = \sum_{i=0}^{5} a_i t^i, \ d(t) = \sum_{i=0}^{5} b_i t^i, \ |\kappa(t)| \le \kappa_{\max}, \\ &v_{\min} \le \sqrt{\dot{s}(t)^2 + \dot{d}(t)^2} \le v_{\max}, \ |\ddot{s}(t)| \le a_{\max}, \\ &|\ddot{d}(t)| \le a_{\text{lat,max}}, \ R(s(t), d(t), t) \le R_{\max}, \forall t \in [0, T], \end{aligned} \tag{11}$$

with the cost function $J(\boldsymbol{\tau})$ given by

$$J(\boldsymbol{\tau}) = w_1 J_{\text{smoothness}} + w_2 J_{\text{risk}} + w_3 J_{\text{efficiency}} + w_4 J_{\text{comfort}}$$
$$J_{\text{smoothness}} = \int_0^T (\dddot{s}(t)^2 + \dddot{d}(t)^2) dt, J_{\text{risk}} = \int_0^T R(s, d, t) dt,$$
$$J_{\text{efficiency}} = T + \eta \int_0^T (v_{des} - \dot{s}(t))^2 dt,$$
$$J_{\text{comfort}} = \int_0^T (\ddot{s}(t)^2 + \ddot{d}(t)^2) dt,$$

where $J_{\text{smoothness}}$ minimizes jerk for smoothness, J_{risk} accounts for risk via DRF, $J_{\text{efficiency}}$ balances travel time and velocity deviation, and J_{comfort} reduces acceleration form improved comfort. T is the trajectory duration, v_{des} the desired velocity, and η a weighting factor. $v_{\min}$ and $v_{\max}$ are the allowable speed limits, $a_{\max}$ the maximum longitudinal acceleration, $a_{\text{lat,max}}$ the maximum lateral acceleration, $\kappa_{\max}$ the maximum curvature, and $R_{\max}$ the maximum risk.

The spatial trajectory generation problem (11) can be solved using a receding horizon approach to obtain the optimal

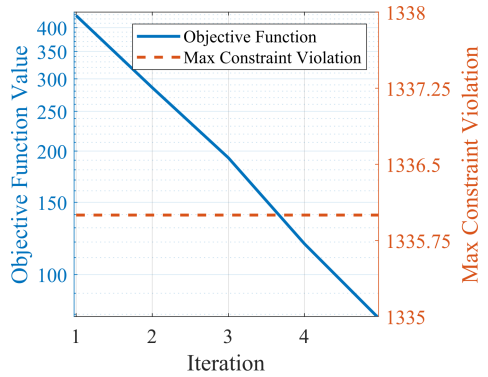


Fig. 4: Convergence analysis of the proposed SQP approach.

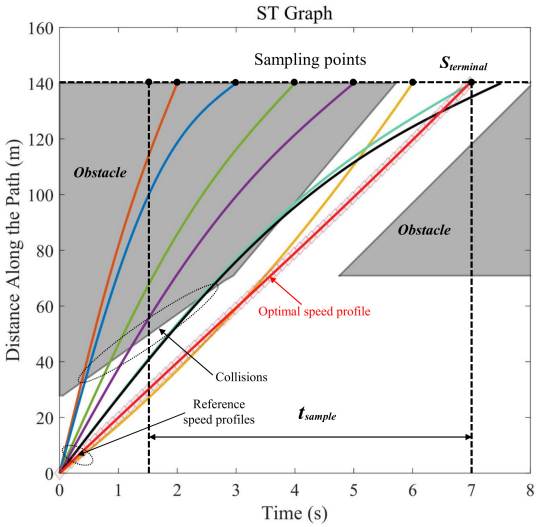


Fig. 5: Constructing the ST Graph.

trajectory $\boldsymbol{\tau}^*$. This method allows for continuous replanning and adaptation to dynamic environments while ensuring efficiency. The optimal trajectory $\boldsymbol{\tau}^*$ is executed for a short duration ΔT before the process is repeated.

The problem (11) is a nonlinear optimization that limits its applicability in real-time vehicle control. To address this, we use Sequential Quadratic Programming (SQP) to solve it in iterative manner (Steps 14 - 26 in Algorithm 1), ensuring efficiency for real-time implementation.

D. SQP-based Trajectory Optimization

This section presents the SQP method for solving problem (11). SQP is well suited for real-time trajectory optimization due to its efficient handling of nonlinear constraints, fast convergence, and warm-start capability. Compared to interior-point methods (less efficient with warm starts), gradient descent (slower convergence), and genetic algorithms (higher computational cost) [37], [38], SQP offers a favorable trade-off between solution quality and computational efficiency, as further validated in Section V. To apply SQP, the constrained optimization problem is reformulated using a Lagrangian function that incorporates the objective and constraints into a unified formulation, enabling second-order updates. The Lagrangian function is defined as

$$\mathcal{L}(\boldsymbol{\tau}, \boldsymbol{\lambda}) = J(\boldsymbol{\tau}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{\tau}),\ \boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]^\top \quad (12)$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, and $g_i(\boldsymbol{\tau})$, $i \in [1, m] \cap \mathbb{Z}$, are the constraint functions from (11).

At each iteration k, we solve the QP subproblem

$$\begin{aligned} &\min_{\mathbf{d}} \frac{1}{2}\mathbf{d}^\top \mathbf{H}_k \mathbf{d} + \nabla J(\boldsymbol{\tau}_k)^\top \mathbf{d} \\ &\text{s.t.} \quad \nabla g_i(\boldsymbol{\tau}_k)^\top \mathbf{d} + g_i(\boldsymbol{\tau}_k) \le 0, \quad i \in [1, m] \end{aligned} \quad (13)$$

where $\mathbf{H}_k$ is an approximation of the Hessian of the Lagrangian, and $\mathbf{d}$ is the search direction.

After solving (13), we update $\mathbf{H}_k$ using the BFGS formula:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^\top}{\mathbf{y}_k^\top \mathbf{s}_k} - \frac{\mathbf{H}_k \mathbf{s}_k \mathbf{s}_k^\top \mathbf{H}_k}{\mathbf{s}_k^\top \mathbf{H}_k \mathbf{s}_k} \quad (14)$$

where $\mathbf{s}_k = \boldsymbol{\tau}_{k+1} - \boldsymbol{\tau}_k$ and $\mathbf{y}_k = \nabla_{\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\tau}_{k+1}, \boldsymbol{\lambda}_{k+1}) - \nabla_{\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\tau}_k, \boldsymbol{\lambda}_k)$. The trajectory is updated using

$$\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k + \alpha_k \mathbf{d}_k \quad (15)$$

where α_k is the step size determined by a line search method to ensure sufficient reduction in the merit function. The Lagrange multipliers are updated using

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k (\boldsymbol{\lambda}_k^* - \boldsymbol{\lambda}_k) \quad (16)$$

where $\boldsymbol{\lambda}_k^*$ is solved from the QP subproblem.

The proposed SQP algorithm iterates until convergence or a maximum iteration limit is reached. To assess its real-time efficiency, we analyze the convergence behavior of the spatial optimization stage. Fig. 4 shows the objective value drops sharply, enabling rapid generation of safe and spatially feasible paths for real-time autonomous driving.

IV. Space-Time Trajectory Optimization

The candidate trajectory generated in Stage I provides a kinematically feasible motion plan in the Frenet frame, combining spatial geometry with an initial time parameterization via quintic polynomials. However, this time allocation is determined only by sampled end states and boundary conditions, without explicitly accounting for time-dependent obstacle interactions. Stage II therefore employs space-time (ST) graphs to refine the velocity profile along the selected path, resolving dynamic obstacle conflicts while enforcing speed and acceleration constraints. This two-stage decomposition separates candidate generation from temporal scheduling, reducing spatiotemporal planning complexity and enabling efficient real-time implementation.

A. ST Graph Construction

The ST graph (see example in Fig. 5) is a two-dimensional representation with the horizontal axis denoting time (t) and the vertical axis representing vehicle's longitudinal position along the reference path (s). The obstacles are shown as grey regions based on their predicted trajectories. Let $\mathcal{O} = \{O_i\}_{i=1}^{M}$

be the set of M obstacles, where O_i contains the longitudinal position s_i predicted over $[0, T_{\max}]$ as follows:

$$s_i(t) = \begin{cases} s_i(0) + v_i(0)t, & 0 \le t < 3 \text{ s} \\ s_i(0) + v_i(0)t + \frac{1}{2}a_i(0)t^2, & 3 \text{ s} \le t < 5 \text{ s} \\ f_{\text{m}}(s_i(0), v_i(0), a_i(0), r\_c, t), & t \ge 5 \text{ s} \end{cases} \quad (17)$$

where $f_{\text{m}}(s_i(0), v_i(0), a_i(0), r\_c, t) = \sum_{j=0}^{n} c_j t^j$ predicts obstacle's future position based on its initial state, road context r_c, and time t. The coefficients c_j are determined through regression analysis of historical trajectory data.

The piecewise prediction model in (17) is designed to balance prediction accuracy and computational efficiency across different horizons. For short-term prediction ($t < 3$ s), a constant-velocity model is adopted, which has been shown to provide accuracy under steady-motion assumptions [39]. For medium-term prediction ($3 \text{ s} \le t < 5$ s), a constant-acceleration (CA) model captures gradual speed variations. For longer horizons ($t \ge 5$ s), a polynomial regression model is used to account for more complex behaviors influenced by road context. This hierarchical design reflects the increasing uncertainty over longer prediction horizons while maintaining real-time computational feasibility.

We acknowledge that the CA model in the medium-term horizon is subject to error accumulation from double integration. This is mitigated by three mechanisms: the CA model is confined to a narrow 2-second window within the piecewise structure (17); the time-dependent buffer $\epsilon(t)$ in (18) compensates for growing prediction uncertainty; and the receding-horizon replanning ($\Delta T = 0.5$ s) continuously corrects predictions with updated measurements, preventing commitment to stale estimates.

In practice, there may exist prediction uncertainty, i.e. the potential deviation of an obstacle's actual future position from its predicted position. To account for this uncertainty, we introduce a time-dependent buffer:

$$\epsilon(t) = \epsilon_0 + k_\epsilon t \quad (18)$$

where ϵ_0 is the initial buffer based on the accuracy of current obstacles state estimation and k_ϵ is a growth rate reflecting increasing uncertainty over time. The value k_ϵ depends on factors like state estimation accuracy (i.e. the precision of sensors and algorithms), obstacle type (e.g. vehicles and pedestrians), and environmental conditions (e.g. weather, road conditions, and traffic density).

By applying (18), we then define the minimum and maximum longitudinal positions of the i-th obstacle as

$$s_{\min,i}(t) = s_i(t) - \epsilon(t), \quad s_{\max,i}(t) = s_i(t) + \epsilon(t). \quad (19)$$

The obstacle region in the ST space is defined as

$$\Omega_{obs} = \cup_{i=1}^{M} \Omega_i(t) \quad (20)$$

where $\Omega_i(t) = \{(s,t) | s_{\min,i}(t) \le s \le s_{\max,i}(t), t \in [0, T_{\max}]\}$ is the ST region occupied by the i-th obstacle at time t.

With (20), we identify the safe corridor $\Omega_{safe}(t)$ that represents areas in the ST space that remain obstacle-free for the HV at time t, which lays the basis for safe trajectory planning. The safe corridor is obtained by

$$\Omega_{safe}(t) = \mathbb{R} \setminus \Omega_{obs}(t) \quad (21)$$

where $\mathbb{R}$ denotes the entire space-time domain. $\Omega_{safe}(t)$ is further discretized into a series of time steps $t_k k = 1^N$, where at each time step t_k, we define the safe longitudinal position range $\Omega_{safe}(t_k) = [s_{\min}(t_k), s_{\max}(t_k)]$.

Algorithm 2 ST-based Trajectory Optimization

Input: Initial trajectory $\tau_0(t)$, obstacle set $\mathcal{O}$, planning horizon $T_{\max}$, time step Δt.
Construct ST graph over $[0, T_{\max}]$ and extract $\Omega_{safe}(t)$.
Reparameterize $\tau_0(t)$ to obtain initial $\boldsymbol{\theta}_0 = [\mathbf{s}_0, \mathbf{d}_0, \mathbf{t}_0]^\top$.
Optimize velocity profile V^* in (23).
$l \leftarrow, 0$ $t_l \leftarrow 0$
while not converged AND $t_l < T_{\max}$ **do**
    Compute $J(\boldsymbol{\theta}_l)$ $\mathbf{g}(\boldsymbol{\theta}_l)$, $\mathbf{h}(\boldsymbol{\theta}_l)$ using (24)
    Solve QP subproblem (13) for $\mathbf{d}_l$.
    Perform line search to determine step size α_l.
    $\boldsymbol{\theta}_{l+1} \leftarrow \boldsymbol{\theta}_l + \alpha_l \mathbf{d}_l$
    Update $\mathbf{H}_{l+1}$ from (14).
    Update $\boldsymbol{\tau}_{l+1}$ from (15).
    Update $\boldsymbol{\lambda}_{l+1}$ from (16).
    $t_l \leftarrow t_l + \Delta t$
    $l \leftarrow l + 1$
end while
Reconstruct continuous trajectory $\tau^*(t)$ from $\boldsymbol{\theta}^*$ and V^* for $t \in [0, T_{\max}]$.
Apply quintic spline interpolation to smooth $\tau^*(t)$.
Return: Optimized trajectory $\tau^*(t)$

B. Reparameterization of Initial Trajectory

To optimize in the ST space, we reparameterize the initial trajectory $\tau_0(t)$ using waypoints and the corresponding time. The discretized trajectory is defined as $\tau = \{(s_i, d_i, t_i) \mid i \in [1, N]\}$, where (s_i, d_i) represents the vehicle's position at the i-th waypoint, t_i is the time between waypoints i and $i+1$, and N is the total number of waypoints. These parameters are organized into a decision vector

$$\boldsymbol{\theta} = [s_1, \ldots, s_N, d_1, \ldots, d_N, t_1, \ldots, t_N]^\top. \quad (22)$$

This $3N \times 1$ vector contains all variables to be optimized, allowing the use of standard optimization techniques. After reparameterizing the trajectory, we impose kinodynamic, safety, and collision avoidance constraints to ensure the trajectory τ is feasible and safe in the ST space. Based on the ST graph (Fig. 5), we derive safe time windows for each waypoint, preventing the vehicle from entering obstacle regions and ensuring collision avoidance through timing adjustments.

C. Integrated Trajectory Optimization

To ensure the trajectory is both spatially feasible and temporally optimized, we perform an integrated optimization (outlined in Algorithm 2), consisting of two main steps:

1) **Velocity profile optimization.** We first optimize the velocity profile along the trajectory. Define the velocity vector $V = [v_1, v_2, \ldots, v_N]^\top$, where v_i is the velocity at the i-th waypoint. The optimal velocity vector V^* is solved from

$$V^* := \arg\min_V \frac{1}{2} V^\top Q V + c^\top V \quad \text{s.t.} \quad A_{ineq} V \le b_{ineq},\ A_{eq} V = b_{eq} \tag{23}$$

with $Q = 2w_2 A_s^\top A_s + 2w_4 A_a^\top A_a + 2w_5 I$ and $c = -2w_5 v_{\text{ref}} \mathbf{1}$, where A_s and A_a are jerk and acceleration matrices. w_2, w_4, w_5 are weighting coefficients. v_{ref} is the reference velocity. $\mathbf{1}$ is a vector of ones. The matrix A_{ineq} includes rows for velocity, acceleration, jerk limits, and ST graph safety constraints, while A_{eq} enforces initial and final velocity conditions.

2) **Full trajectory refinement.** Based on the optimized velocity profile V^*, we then refine the full trajectory $\boldsymbol{\theta} = [\mathbf{s}, \mathbf{d}, \mathbf{t}]^\top$ by solving the optimization problem:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \quad \text{s.t.} \quad \mathbf{g}(\boldsymbol{\theta}) \le \mathbf{0},\ \mathbf{h}(\boldsymbol{\theta}) = \mathbf{0} \tag{24}$$

where $\mathbf{g}(\boldsymbol{\theta}) \le 0$ represents the set of inequality constraints:

$$\begin{aligned}
&v_{\min} \le (s_{k+1} - s_k)/t_k \le v_{\max},\ k \in [1, N-1],\\
&2|(s_{k+1} - s_k)/t_k - (s_k - s_{k-1})/t_{k-1}|/(t_k + t_{k-1}) \le a_{\max},\\
&|a_{k+1} - a_k|/t_k \le j_{\max},\ k \in [2, N-2],\\
&s_{\min}(t_k) \le s_k \le s_{\max}(t_k), t_k^{\min} \le t_k \le t_k^{\max}, k \in [1, N].
\end{aligned}$$

$\mathbf{h}(\boldsymbol{\theta}) = \mathbf{0}$ represents the set of equality constraints: $s_1 = s_0(0), d_1 = d_0(0), s_N = s_f, d_N = d_f$. The cost function $J(\boldsymbol{\theta}) = w_1 J_{\text{time}} + w_2 J_{\text{smoothness}} + w_3 J_{\text{safety}} + w_4 J_{\text{comfort}} + w_5 J_{\text{efficiency}}$ consists of five parts:

$$\begin{aligned}
&J_{\text{time}} = \sum_{k=1}^{N} t_k,\ J_{\text{smoothness}} = \sum_{k=2}^{N-1} ((a_{k+1} - a_k)/t_k)^2,\\
&J_{\text{safety}} = \sum_{k=1}^{N} \exp(-\alpha \cdot \min(s_k - s_{\min}(t_k), s_{\max}(t_k) - s_k)),\\
&J_{\text{comfort}} = \sum_{k=2}^{N-1} a_k^2,\ J_{\text{efficiency}} = \sum_{k=1}^{N-1} (v_k - v_{\text{ref}})^2.
\end{aligned}$$

Here α is a scaling factor, and (s_k, v_k) denote the longitudinal position and velocity at waypoint k (from V^*). a_k and j_k are the acceleration and jerk, bounded by $a_{\max}$ and $j_{\max}$. $(s_0(0), d_0(0))$ and (s_f, d_f) is the initial and terminal states, while $[t_i^{\min}, t_i^{\max}]$ is the safe time window at waypoint i.

We note that the safety constraints in (24) are consistent with (11), as the corridor bounds $[s_{\min}(t_k), s_{\max}(t_k)]$ are derived from the same obstacle predictions and uncertainty buffer $\epsilon(t)$. Since Stage II starts from a Stage I feasible trajectory and SQP enforces $\mathbf{g}(\boldsymbol{\theta}) \le 0$ at each iteration, safety is preserved throughout refinement.

The complete optimization procedure is summarized in Algorithm 2. Problem (24) is solved via SQP following Section III-D, extended to incorporate spatiotemporal constraints. As shown in Fig. 6, SQP converges rapidly within the first 20 iterations, followed by temporal adjustment and final fine-tuning up to 100 iterations, producing smooth velocity profiles and efficient time allocation. This demonstrates that our framework combines fast spatial feasibility with effective temporal refinement for real-time autonomous driving.

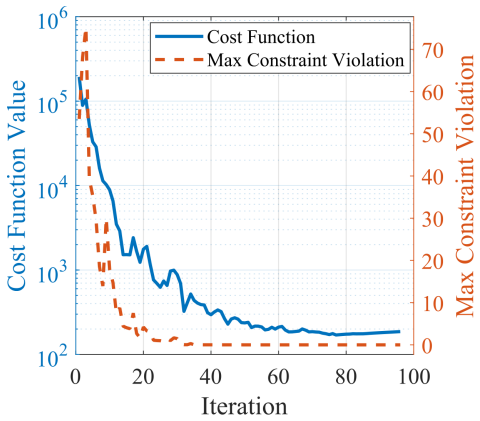


Fig. 6: Convergence analysis of the proposed SQP method.

TABLE I: KEY SIMULATION PARAMETERS

Parameter	Value	Description
Vehicle & Planning		
L	2.5 m	Vehicle wheelbase
$l_v,\ w_v$	4.5 m, 1.8 m	Vehicle length and width
$v_{\max}$	20 m/s	Maximum velocity
$a_{\max},\ a_{\text{lat,max}}$	3, 2 m/s^2	Max. longitudinal / lateral accel.
$T_{\max},\ \Delta T$	8 s, 0.5 s	Planning horizon / replanning interval
Dynamic Risk Field		
A_i	1.0	Static risk field amplitude
$\sigma_{s,i},\ \sigma_{d,i}$	$1.5l_v,\ 1.2w_v$	Static field spread (longitudinal / lateral)
B_j	0.8	Dynamic risk field amplitude
$\sigma_0,\ k_v$	0.65 m, 0.4	Baseline spread / velocity scaling in (7)
$k_0,\ k_r$	2.0, 0.3	Baseline sensitivity / velocity scaling
$\alpha,\ L_j$	1.5, l_v	Safety factor / obstacle length
Prediction Uncertainty		
$\epsilon_0,\ k_\epsilon$	0.5 m, 0.1 m/s	Initial buffer / buffer growth rate

D. Trajectory Reconstruction and Smoothing

After obtaining the V^* and optimized waypoints, we reconstruct the full trajectory in the Frenet frame using

$$s(t) = s_i + v_i^*(t - t_i),\ d(t) = d_i + \frac{d_{i+1} - d_i}{t_{i+1} - t_i}(t - t_i), t \in [t_i, t_{i+1}].$$

The quintic polynomials in (9) are applied to interpolate between consecutive waypoints as follows:

$$s(t) = \sum_{j=0}^{5} a_j (t - t_i)^j, d(t) = \sum_{j=0}^{5} b_j (t - t_i)^j, t \in [t_i, t_{i+1}],$$

where the coefficients $\{a_j\}_{j=0}^5$ and $\{b_j\}_{j=0}^5$ for each segment $[t_i, t_{i+1}]$ are determined by solving equations that enforce continuity up to the third derivative at each waypoint, ensuring smooth transitions. By integrating spatiotemporal optimization, the obtained trajectories ensure collision avoidance whilst maintaining smooth and feasible vehicle dynamics.

V. SIMULATION RESULTS

We evaluate the proposed method in three MATLAB 2023b simulation cases, focusing on both the two-stage spatiotemporal planning framework and the SQP solver in Stage II. Solver performance is compared against DE [40], PSO

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3680524

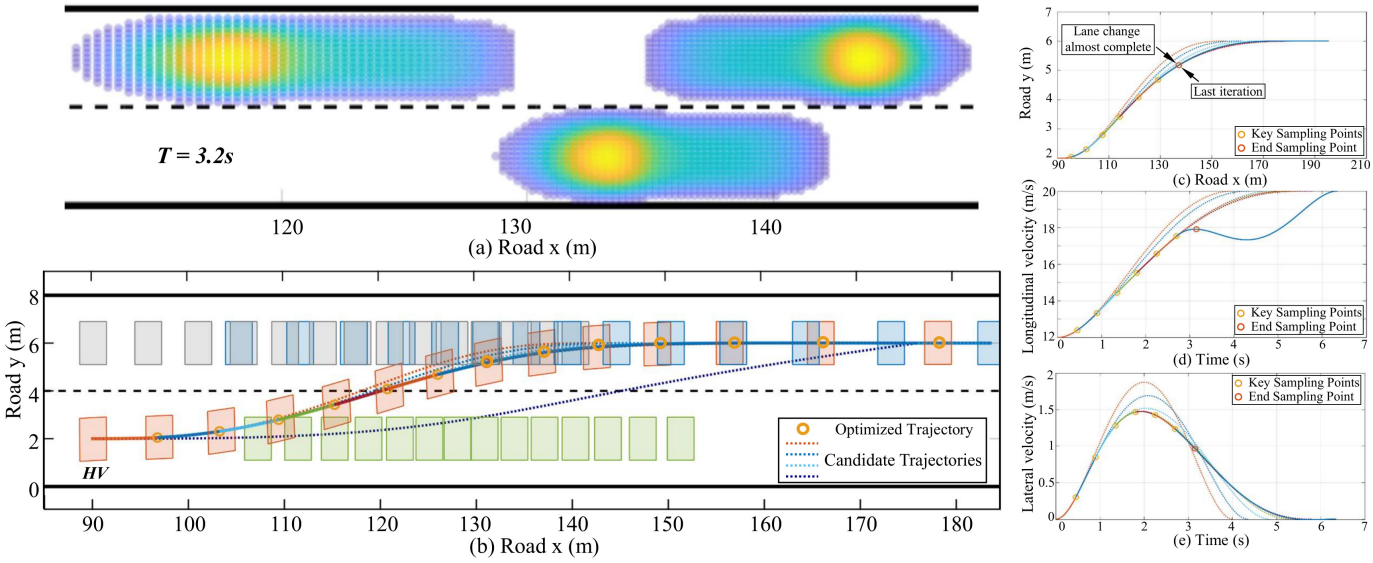


Fig. 7: Decision-making and initial planning in Case 1. The dotted lines represent candidate trajectories, solid lines denote optimized trajectories, and colored hollow points indicate nodes from segmented receding optimization. (a) DRF constructed based on uncertainty and time-dependent buffer $\epsilon(t)$. (b) Candidate and optimized trajectories from initial planning. (c) and (d) Longitudinal and lateral velocity variations of candidate and optimized trajectories over time, respectively.

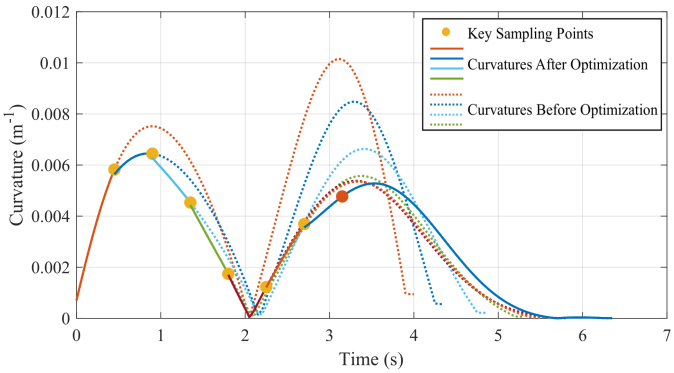


Fig. 8: Curvatures during receding optimization in Case 1.

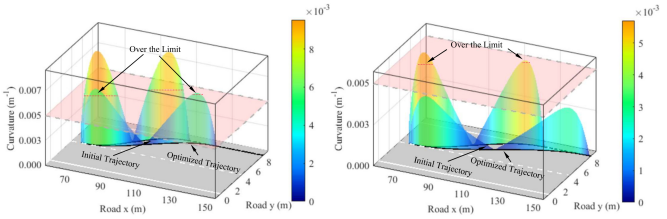


Fig. 9: Comparison of curvature variations before and after optimization for Case 1. The dashed line denotes the initial trajectory and the solid line denotes the optimized trajectory. The shaded region indicates curvature constraint violations. (a) After Stage I spatial optimization. (b) After Stage I and Stage II temporal optimization, with no violations observed.

[41], GA [37], I_P [42], and P_S [43]. Spatial–temporal decomposition is validated via stage-wise comparisons and trajectory analysis. Key parameters are given in Table I.

A. Case 1: Lane Changing on Two-lane Highway

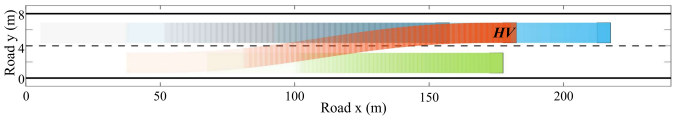
This simulation considers an interactive driving scenario where the HV starts in the lower lane behind a front vehicle, while two SVs are in the upper lane. The HV's task is to change lane to the upper lane while interacting with the SVs.

Figure 7 illustrates the decision-making and initial planning by using our approach. Fig. 7(a) shows the DRF at $T = 3.2$ s, indicating potential conflict areas, with yellow for higher risk and blue for lower risk. Fig. 7(b) displays the road layout with gray candidate trajectories and the optimized HV trajectory marked with a solid line and circular markers. The red rectangles represent the HV's position. Fig. 7(c)–(e) show kinematic graphs of the vehicle's lateral and longitudinal positions, and speed changes during lane changing. The optimized trajectory is smoother than the initial candidate trajectories, ensuring driving comfort and safety.

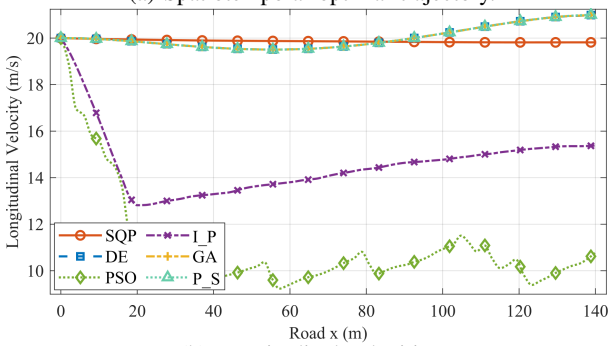
Figure 8 shows the curvature variations of candidate and optimized trajectories during lane changing. All trajectories exhibit two peaks, with the first at around 1 s second and the second between 3-4 s. The optimized trajectory has a lower second peak that occurs later and a smoother decline, indicating a smoother, safer lane-change compared to the more abrupt, aggressive candidate trajectories.

Figure 9 shows the impact of our two-stage optimization on trajectory curvature. In Fig. 9(a), after the first-stage spatial optimization, the curvature often exceeds the defined limit, indicated by peaks crossing the yellow plane. Although the optimized trajectory reduces these peaks, some areas still approach the limit. In Fig. 9(b), the second-stage temporal optimization further reduces curvature, ensuring no sections exceed the limit. This makes the trajectory smoother without sacrificing efficiency, enhancing both safety and comfort. These results demonstrate that the second-stage temporal optimization plays

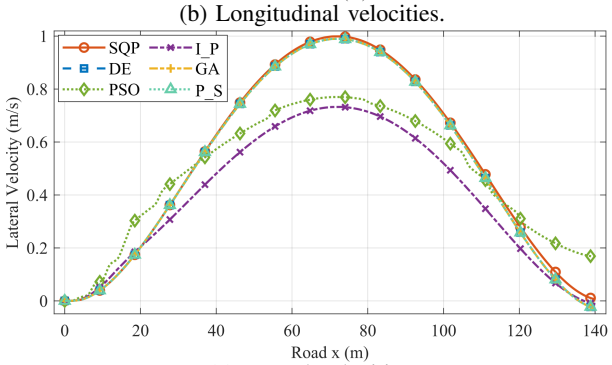
This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3680524



(a) Spatiotemporal optimal trajectory.

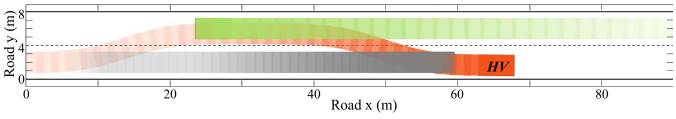


(b) Longitudinal velocities.

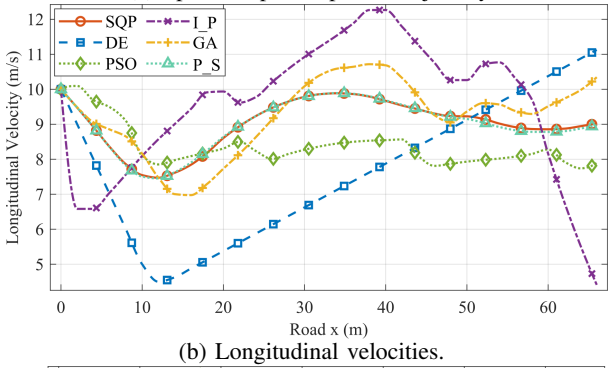


(c) Lateral velocities.

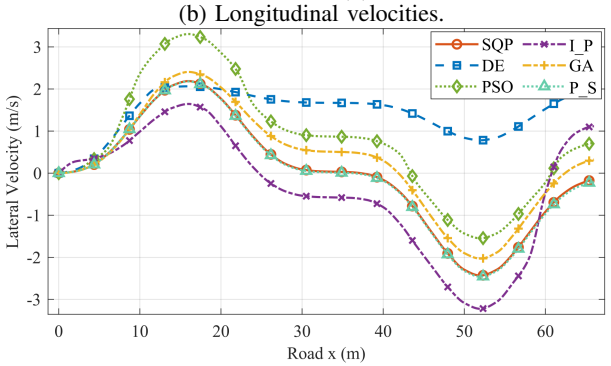
Fig. 10: Spatiotemporal trajectory optimization in Case 1.



(a) Spatiotemporal optimal trajectory.



(b) Longitudinal velocities.



(c) Lateral velocities.

Fig. 11: Spatiotemporal trajectory optimization in Case 2.

TABLE II:
COMPARISON OF ALGORITHM PERFORMANCES IN CASE 1

Methods	x_a (m/s^2)	y_a (m/s^2)	x_j (m/s^3)	y_j (m/s^3)	T (s)
DE	0.29	0.29	1.39	1.42	7.04 ± 0.04
PSO	1.27	0.08	2.40	0.19	16.63 ± 0.08
I_P	0.91	0.15	3.03	0.51	–
GA	0.28	0.29	1.40	1.42	7.04 ± 0.03
P_S	0.28	0.29	1.41	1.42	7.05 ± 0.05
Ours	**0.03**	**0.28**	**0.13**	**1.39**	$\mathbf{7.02 \pm 0.02}$

x_a: average longitudinal acceleration; y_a: average lateral acceleration; x_j: average longitudinal jerk; y_j: average lateral jerk; T: average lane change time; –: a collision occurred.

a critical role in resolving residual geometric infeasibility left by spatial planning alone, highlighting the necessity of the proposed two-stage spatiotemporal framework.

Fig. 10(a) illustrates the optimal trajectory planned by our method, with the HV (in red) changing lanes from left to right and SVs (colored rectangles). The vehicle's trajectory history fades over time. In Fig. 10(b), the proposed SQP method produces a stable longitudinal velocity profile with minimal oscillations, outperforming the D_E and AS methods, which exhibit larger variations. Similarly, Fig. 10(c) shows that the SQP maintains a smooth lateral velocity profile, while PSO and GA demonstrate noticeable fluctuations, particularly in lateral motion. Table II further compares our method against benchmark algorithms in terms of average longitudinal and lateral accelerations (x_a, y_a), jerk (x_j, y_j), and lane change time (T). Our method excels across all metrics, achieving the lowest accelerations, jerk, and lane change time ($\mathbf{7.02 \pm 0.02}$ s), highlighting its superior smoothness and efficiency.

B. Case 2: Overtaking on A Two-lane, Two-way Road

Compared to Case 1, Case 2 considers an even more challenging scenario of lane changing in two-lane highway. The HV starts in the lower lane behind an SV, while another SV in the upper lane drives in the opposite direction. The HV must change lanes twice to avoid collisions with these SVs.

Figure 11 illustrates the results of trajectory planning. Fig. 11(a) presents the final optimal trajectory, showing the HV navigating around the SVs over a distance of 90 m. The trajectory depicts a safe lane changing maneuver from the lower lane to the upper lane, and back to the lower lane. Fig. 11(b) displays the longitudinal velocity profile along the path, with the proposed SQP approach showing minimal oscillations compared to other methods. Fig. 11(c) shows the lateral velocity profile, highlighting the smoothness of the lane changing maneuver. Results indicates sharp jumps and points during lane changing, where the proposed SQP achieves better stability than other methods.

The ST graph in Fig. 12 shows paths generated by different optimization methods. The optimal path by SQP effectively avoids collisions, with a consistent slope indicating smooth progress and balanced speed variations for comfort and efficiency. In contrast, PSO shows early collisions due to

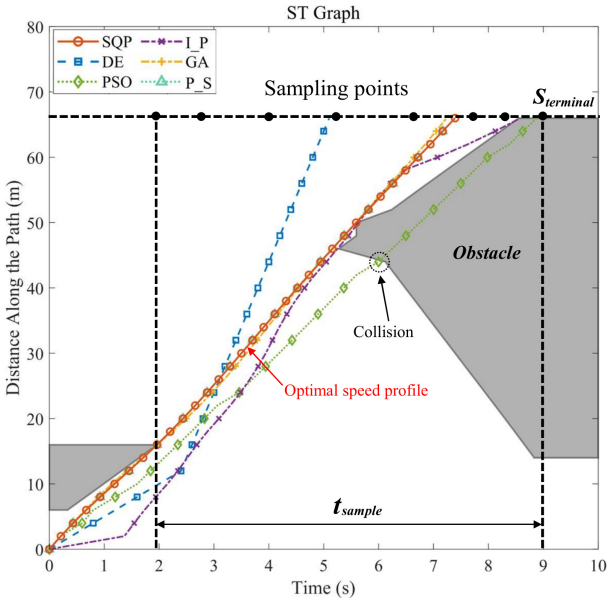


Fig. 12: Comparison of optimal paths in Case 2.

TABLE III:
COMPARISON OF ALGORITHM PERFORMANCES IN CASE 2

Methods	x_a (m/s^2)	y_a (m/s^2)	x_j (m/s^3)	y_j (m/s^3)	T (s)
DE	2.42	**0.93**	9.94	3.82	**5.09** $\pm$ 0.04
PSO	0.68	1.11	1.06	1.93	–
I_P	2.23	1.34	5.411	3.15	7.39 $\pm$ 0.04
GA	1.45	1.26	3.02	2.76	6.84 $\pm$ 0.03
P_S	0.81	1.22	1.72	2.64	7.39 $\pm$ 0.05
Ours	**0.78**	1.22	**1.68**	**2.63**	7.38 $\pm$ **0.02**

x_a: average longitudinal acceleration; y_a: average lateral acceleration; x_j: average longitudinal jerk; y_j: average lateral jerk; T: average lane change time; $-$: a collision occurred.

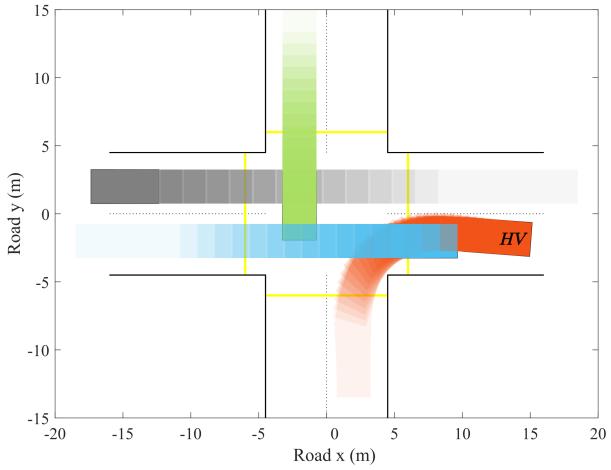
suboptimal path, while the proposed SQP reaches the terminal state S_{terminal} smoothly.

In Table III, we compare the performance of our method against benchmark algorithms. The proposed method achieves the lowest longitudinal acceleration (**0.78** m/s^2) and jerk values, indicating smooth and efficient maneuvers. DE yields the shortest lane change time (**5.09** s), while the proposed method closely follows at $7.38 \pm \mathbf{0.02}$ s.

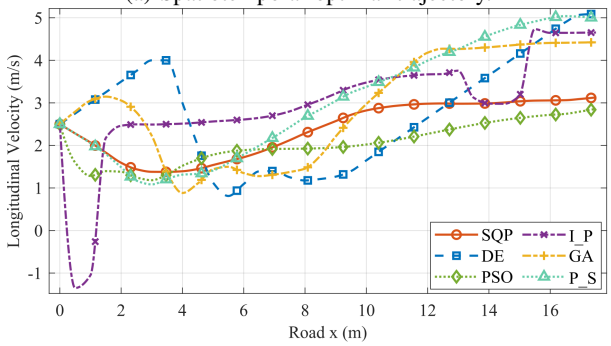
C. Case 3: Turning at Four-port Intersection

This case considers a four-port intersection with 2.5 m lane width, three SVs and one HV. The HV starts in the lower port, with SVs approaching from different directions, requiring the HV to avoid all SVs while changing lanes at the intersection.

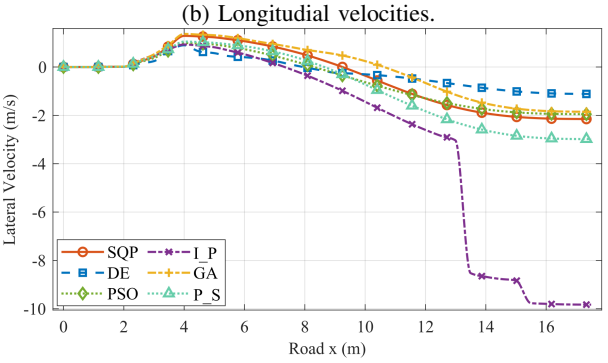
Figure 13 illustrates the results of the proposed trajectory planning. Fig. 13(a) presents the final optimal trajectory, where the HV (in orange) successfully navigates through the intersection while avoiding SVs, taking a slight right turn before proceeding straight. In Fig. 13(b), noticeable oscillations in the DE and GA methods indicate less stable velocity control, which could lead to discomfort and potential safety risks. The proposed SQP method shows minimal velocity fluctuations, demonstrating superior longitudinal control, which is essential



(a) Spatiotemporal optimal trajectory.



(b) Longitudial velocities.



(c) Lateral velocities.

Fig. 13: Spatiotemporal trajectory planning in Case 3.

for ensuring both smooth acceleration and deceleration during dynamic maneuvers. Figure 13(c) presents the lateral velocity profile, where noticeable spikes in the DE and A_S methods suggest sudden lateral movements, which can lead to reduced passenger comfort and potential instability. The proposed SQP trajectory, with minimal lateral velocity variations, demonstrates better lateral control, which is crucial for executing safe and comfortable lane turns in intersection scenarios.

Figure 14 illustrates the lateral acceleration profile, where the SQP method maintains minimal lateral acceleration, ensuring stability and comfort during the maneuver. In contrast, the DE method exhibits fluctuations, indicating abrupt lateral forces that could compromise vehicle stability and passenger comfort. Maintaining low lateral acceleration is essential for minimizing the impact of sudden maneuvers and providing a comfortable driving experience.

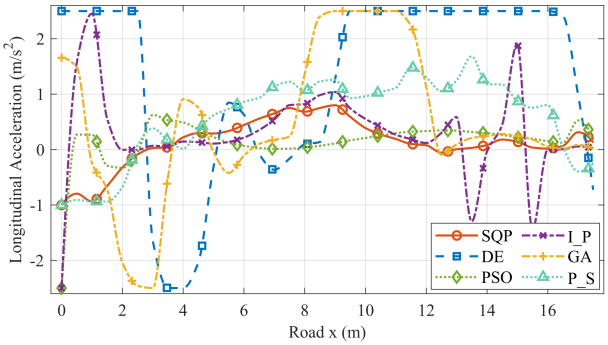


Fig. 14: Comparison of longtudinal accelerations in Case 3.

TABLE IV:
COMPARISON OF ALGORITHM PERFORMANCES IN CASE 3

Methods	x_a (m/s^2)	y_a (m/s^2)	x_j (m/s^3)	y_j (m/s^3)	T (s)
DE	1.85	**0.46**	7.64	1.77	$\mathbf{7.10} \pm 0.04$
PSO	0.30	0.43	0.53	0.87	–
I_P	0.58	0.98	0.94	2.85	14.89 ± 0.05
GA	0.99	0.77	2.89	2.65	$\mathbf{7.23 \pm 0.03}$
P_S	0.86	0.89	2.75	3.10	7.17 ± 0.05
Ours	**0.34**	**0.63**	**0.73**	**1.57**	7.92 ± 0.04

x_a: average longitudinal acceleration; y_a: average lateral acceleration; x_j: average longitudinal jerk; y_j: average lateral jerk; T: average lane change time; $-$: a collision occurred.

Table IV shows that the proposed method achieves comparative performance, with the lowest lateral acceleration (0.63 m/s^2) and jerk values, indicating smoother and more controlled maneuvers. The DE method achieves the shortest lane change time (7.10 s), while the proposed method maintains a reasonable time of 7.92 ± 0.04 s.

VI. CONCLUSION

This paper presents a two-stage trajectory planning framework for AVs in dynamic environments. The contributions include a DRF for unified representation of static and dynamic obstacles, a spatial–temporal decomposition combining trajectory generation and temporal optimization, an efficient SQP-based solver, and a multi-objective cost balancing safety, efficiency, and comfort. A receding horizon strategy enables continuous replanning. Simulation results across three scenarios demonstrate the effectiveness of the framework in generating safe, efficient, and comfortable trajectories. The optimization-based formulation offers strong interpretability and enables explicit safety constraints, achieving consistent performance across scenarios without task-specific retraining—properties critical for safety-critical deployment. Optimization-based and learning-based approaches are complementary in practice: the latter excels in perception and prediction, while the former ensures reliable constraint satisfaction and safety fallback. Limitations include reliance on accurate obstacle prediction and validation in relatively sparse scenarios (up to four vehicles). Future work will focus on large-scale validation (e.g., CARLA and real-world settings), integration of learning-based prediction and adaptive tuning under uncertainty, and improved scalability and efficiency in dense traffic.

REFERENCES

[1] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, *et al.*, "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, no. 113816, p. 113816, 2021.

[2] J. Pérez, V. Milanés, T. De Pedro, and L. Vlacic, "Autonomous driving manoeuvres in urban road traffic environment: A study on roundabouts," in *Proc. IFAC World Congress*, vol. 44, 2011, pp. 13795–13800.

[3] M. Guériau and I. Dusparic, "Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic," in *Proc. ITSC*, 2020, pp. 1–8.

[4] S. Narayanan, E. Chaniotakis, and C. Antoniou, "Factors affecting traffic flow efficiency implications of connected and autonomous vehicles: A review and policy recommendations," *Adv. Transp. Policy Plan.*, vol. 5, pp. 1–50, 2020.

[5] P. Scheffe, M. V. A. Pedrosa, K. Flaßkamp, and B. Alrifaee, "Receding horizon control using graph search for multi-agent trajectory planning," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 3, pp. 1092–1105, 2023.

[6] J. Hu, Y. Zhang, and S. Rakheja, "Adaptive lane change trajectory planning scheme for autonomous vehicles under various road frictions and vehicle speeds," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1252–1265, 2023.

[7] G. Chen, Y. Zhang, and X. Li, "Attention-based highway safety planner for autonomous driving via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 162–175, 2024.

[8] G. Chen, T. Wu, X. Li, and Y. Zhang, "Secure and safe control of connected and automated vehicles against false data injection attacks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12347–12360, 2024.

[9] J. Zhu, K. Gao, H. Li, Z. He, and C. O. Monreal, "Bi-level ramp merging coordination for dense mixed traffic conditions," *Fundam. Res.*, vol. 4, no. 5, pp. 992–1008, 2024.

[10] H. Wang, H. Gao, S. Yuan, H. Zhao, *et al.*, "Interpretable decision-making for autonomous vehicles at highway on-ramps with latent space reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8707–8719, 2021.

[11] M. Pourabdollah, E. Bjärkvik, F. Fürer, B. Lindenberg, and K. Burgdorf, "Calibration and evaluation of car following models using real-world driving data," in *Proc. ITSC*, 2017, pp. 1–6.

[12] X. Hu, Z. Zheng, D. Chen, and J. Sun, "Autonomous vehicle's impact on traffic: Empirical evidence from waymo open dataset and implications from modelling," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6711–6724, 2023.

[13] D. Li, J. Zhang, and G. Liu, "Autonomous driving decision algorithm for complex multi-vehicle interactions: An efficient approach based on global sorting and local gaming," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6927–6937, 2024.

[14] Z. Hong, Q. Lin, and B. Hu, "Knowledge distillation-based edge-decision hierarchies for interactive behavior-aware planning in autonomous driving system," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11040–11057, 2024.

[15] W. Ding, L. Zhang, J. Chen, and S. Shen, "Epsilon: An efficient planning system for automated vehicles in highly interactive environments," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1118–1138, 2022.

[16] M. Lv, Y. Li, H. Liang, B. Sun, C. Yang, and W. Gui, "A spatial–temporal variational graph attention autoencoder using interactive information for fault detection in complex industrial processes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 3062–3076, 2024.

[17] K. Liu *et al.*, "Optimal time trajectory generation and tracking control for over-actuated multirotors with large-angle maneuvering capability," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8339–8346, 2022.

[18] C. Zhao *et al.*, "A novel direct trajectory planning approach based on generative adversarial networks and rapidly-exploring random tree," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17 910–17 921, 2022.

[19] C.-K. Ho and C.-T. King, "Lac-rrt: Constrained rapidly-exploring random tree with configuration transfer models for motion planning," *IEEE Access*, vol. 11, pp. 97654–97663, 2023.

[20] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," in *Proc. ISVD*, 2007, pp. 38–47.

[21] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized voronoi diagram-based efficient heuristic path planning method for rrts in mobile robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 4926–4937, 2022.

[22] Y. Gao, D. Li, Z. Sui, and Y. Tian, "Trajectory planning and tracking control of autonomous vehicles based on improved artificial potential field," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12468–12483, 2024.

[23] R. Szczepanski, "Safe artificial potential field: Novel local path planning algorithm maintaining safe distance from obstacles," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4823–4830, 2023.

[24] W. Zhong, X. Huang, Y. Wu, R. Yu, and J. Kang, "Decentralized energy management for wireless power transfer assisted platoon autonomous driving: A leader-to-follower approach," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 4, pp. 2073–2083, 2022.

[25] D. Lenz, T. Kessler, and A. Knoll, "Tactical cooperative planning for autonomous highway driving using monte-carlo tree search," in *Proc. IV*, 2016, pp. 447–453.

[26] F. Altché and A. de La Fortelle, "Partitioning of the free space-time for on-road navigation of autonomous ground vehicles," in *Proc. CDC*, 2017, pp. 2126–2133.

[27] Z. Lin, J. Lan, A.-T. Nguyen, and D. Flynn, "Contingency-aware spatiotemporal optimization for safe autonomous vehicle trajectory planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 11, pp. 18 487–18 499, Nov. 2025.

[28] J. Wurts, J. L. Stein, and T. Ersal, "Design for real-time nonlinear model predictive control with application to collision imminent steering," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 6, pp. 2450–2465, 2022.

[29] J. Zhang, S.-C. Chai, B.-H. Zhang, and G.-P. Liu, "Distributed model-free sliding-mode predictive control of discrete-time second-order nonlinear multiagent systems with delays," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12 403–12 413, 2022.

[30] F. Bayer, M. Bürger, and F. Allgöwer, "Discrete-time incremental iss: A framework for robust nmpc," in *Proc. ECC*, 2013, pp. 2068–2073.

[31] Z. Lin, J. Lan, C. Anagnostopoulos, Z. Tian, and D. Flynn, "Safety-critical multi-agent mcts for mixed traffic coordination at unsignalized intersections," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2025.

[32] M. Wang *et al.*, "Speed planning for autonomous driving in dynamic urban driving scenarios," in *Proc. ECCE*, 2020, pp. 1462–1468.

[33] H. Chi, P. Cai, D. Fu, J. Zhai, Y. Zeng, and B. Shi, "Spatiotemporal-restricted A* algorithm as a support for lane-free traffic at intersections with mixed flows," *Green Energy and Intelligent Transportation*, vol. 3, no. 2, p. 100159, 2024.

[34] A. Irshayyid, J. Chen, and G. Xiong, "A review on reinforcement learning-based highway autonomous vehicle control," *Green Energy and Intelligent Transportation*, vol. 3, no. 4, p. 100156, 2024.

[35] Z. Lin, J. Lan, and X. Zhao, "Kan-lstm enhanced multi-agent advantage actor-critic reinforcement learning for autonomous ramp merging," *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2025.

[36] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.

[37] J. Tang *et al.*, "A data-driven timetable optimization of urban bus line based on multi-objective genetic algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2417–2429, 2021.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

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

[40] H. Lin, Y. Han, W. Cai, and B. Jin, "Traffic signal optimization based on fuzzy control and differential evolution algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8555–8566, 2023.

[41] P. Shi and W. Peng, "Preparation of papers for ifac conferences & symposia: Research on model predictive control of lane keeping based on particle swarm optimization," in *Proc. CVCI*, 2021, pp. 1–9.

[42] R. Hult, M. Zanon, S. Gros, and P. Falcone, "A semidistributed interior point algorithm for optimal coordination of automated vehicles at intersections," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 5, pp. 1977–1989, 2022.

[43] F. Wahid *et al.*, "An enhanced firefly algorithm using pattern search for solving optimization problems," *IEEE Access*, vol. 8, pp. 148 264–148 288, 2020.

Zhihao Lin received the M.S. degree in Circuits and Systems from Jilin University, China, in 2022. He is currently pursuing the Ph.D. degree with the James Watt School of Engineering, University of Glasgow, U.K. His research focuses on structure-aware reinforcement learning, multi-agent decision making, and safety-critical planning.

Zhen Tian received the B.Eng. degree in Electronic and Electrical Engineering from the University of Strathclyde, Glasgow, U.K., in 2020, and the Ph.D. degree from the College of Science and Engineering, University of Glasgow, Glasgow, U.K., in 2025. He is currently a postdoctoral researcher at the University of Glasgow. His research interests include interactive agent decision systems and autonomous racing strategies.

Xianxian Zhao received the Ph.D. degree in Electrical Engineering from the University of Birmingham in 2018. She is currently the Principle Investigator of the SEAI-RD&D Programme at University College Dublin. Her research focuses on modelling and control of electric machines and converters, stability and power quality of power-electronic-based power systems, and model order reduction of large power systems.

Hanyang Zhuang (Member, IEEE) received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2018. He was a Postdoctoral Researcher with Shanghai Jiao Tong University from 2020 to 2022. He is currently an Assistant Research Professor with Shanghai Jiao Tong University implementing research works related to intelligent vehicles. His research interest include AD/ADAS system design, high-precision localization, environment perception, and cooperative driving.

Ming Yang received his Master's and Ph.D. degrees from Tsinghua University, Beijing, China, in1999 and 2003, respectively. Presently, he holds the position of Distinguished Professor at Shanghai Jiao Tong University, also serving as the Director of the Innovation Center of Intelligent Connected Vehicles. Dr. Yang has been engaged in the research of intelligent vehicles for more than 25 years.

Jianglin Lan received the Ph.D. degree from the University of Hull in 2017. He has been a Leverhulme Early Career Fellow and Lecturer at the University of Glasgow since 2022. He was a Visiting Professor at the Robotics Institute, Carnegie Mellon University, in 2023. From 2017 to 2022, he held postdoc positions at Imperial College London, Loughborough University, and University of Sheffield. He is an Editor of the International Journal of Adaptive Control and Signal Processing.