

Article

Distributed Multi-Vehicle Cooperative Trajectory Planning and Control for Ramp Merging and Diverging Based on Deep Neural Networks and MPC

Linhua Nie ¹, Tingyang Zhang ^{3,4}, Yunqing Zhao ^{2,*}, Yaqiu Li ⁵, Haoran Li ^{1,2,3,6,*} and Junru Yang ⁶

¹ School of Transportation, Southeast University, Nanjing 211189, China; 213232318@seu.edu.cn

² Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing 100091, China

³ School of Automotive and Traffic Engineering, Wuhan University of Science and Technology, Wuhan 430065, China; 15937493953@163.com

⁴ YTO Group Corporation, Luoyang 71003, China

⁵ Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima 739-8529, Japan; lyq@hiroshima-u.ac.jp

⁶ Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215134, China; yangjunru@tsari.tsinghua.edu.cn

* Correspondence: 24120886@bjtu.edu.cn (Y.Z.); 103200214@seu.edu.cn (H.L.)

Abstract

With the deep integration of the modern automotive industry and artificial intelligence technologies, connected and automated vehicles (CAVs) have emerged as a key breakthrough for improving traffic safety and operational efficiency. This study proposes a distributed multi-vehicle cooperative trajectory planning and control framework for ramp merging and diverging scenarios, integrating Deep Neural Networks (DNNs) with Model Predictive Control (MPC). The methodology consists of three key components: First, a distributed cooperative architecture based on dynamic topology is constructed to effectively reduce communication loads; second, a feature point-based Cubic Bézier Curve trajectory generation method is proposed, enabling flexible path planning with reduced reliance on high-precision maps; finally, a DNN-accelerated MPC solving strategy (NN-MPC) is designed. This strategy employs an offline-trained deep neural network to approximate the online optimization process, supplemented by a terminal Safety Check mechanism and a dynamic surrounding vehicle selection algorithm. Experimental results demonstrate that the proposed method successfully reproduces the planning capability of offline high-precision MPC in ramp merging and diverging scenarios while reducing computation time to the millisecond level. It effectively overcomes the myopic decision-making problem of traditional real-time algorithms, achieving smoother conflict resolution and higher traffic efficiency. Notably, quantitative validation confirms that this cooperative framework achieves an approximate 30% reduction in average travel delay compared to the non-cooperative baseline. This study confirms the engineering advantages of the hybrid architecture under dynamic high-density traffic flows, significantly enhancing the system's real-time response capability while balancing the safety and riding comfort of cooperative driving.

Keywords: trajectory generation; Cubic Bézier Curve; high-density traffic flows; intelligent transportation systems

Academic Editor: Zheng Chen

Received: 2 January 2026

Revised: 15 February 2026

Accepted: 17 February 2026

Published: 25 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the deep integration of the modern automotive industry and artificial intelligence technologies, connected and automated vehicles (CAVs) have emerged as a key breakthrough for improving traffic safety and operational efficiency [1]. As an integral component of intelligent transportation systems (ITS), autonomous driving technology is not limited to single-vehicle autonomous navigation but also aims to achieve full-scale collaboration among vehicles, roads, and cloud systems through Vehicle-to-Everything (V2X) technology. However, despite the significant potential of intelligent vehicles under regular traffic conditions, their development still faces severe challenges in complex bottleneck segments such as freeway ramp merging areas—one of the key scenarios targeted for intelligent transformation in national industrial plans [2].

Ramp merging and diverging areas serve as the bottleneck of traffic flow, involving a series of complex operations including lane changing, acceleration, and merging. Consequently, these areas are high-risk zones for traffic accidents [3]. In real-world traffic environments, ramp merging areas exhibit high dynamics and uncertainty [4]. Traditional single-vehicle intelligence, when addressing such scenarios, is often constrained by limited perception range, making it difficult to obtain global traffic situational awareness. This can easily lead to locally optimal solutions or even deadlock phenomena due to the “bounded rationality” of drivers or algorithms [5], failing to maximize operational efficiency while ensuring safety. For instance, during peak traffic volumes, ramp vehicles often face insufficient merging space, while mainline vehicles may trigger conflicts due to the lack of evasive mechanisms [6]. Standard rule-based methods (e.g., First-Come-First-Served) often fail to allocate gaps efficiently under saturated traffic density (e.g., near-capacity flow rates), leading to stop-and-go waves or deadlocks. This study specifically targets these critical bottleneck scenarios where cooperative efficiency is paramount.

To address these issues, multi-vehicle cooperative control based on V2X technology has become a research hotspot [7,8]. Through Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Infrastructure (V2I) collaboration, vehicles can share state information and make cooperative decisions. Current cooperative control architectures are mainly categorized into centralized and distributed types. While centralized control can achieve global optimality, it suffers from heavy communication loads, exponential growth in computational complexity, and single-point failure risks when dealing with large-scale traffic flows. These limitations make it difficult to meet the stringent real-time requirements of ramp merging scenarios [6]. In contrast, distributed control architectures offer superior robustness and scalability, aligning better with the dynamic characteristics of real traffic flows.

At the control algorithm level, Model Predictive Control (MPC) is widely applied in trajectory planning and control due to its ability to explicitly handle vehicle dynamic constraints, collision avoidance constraints, and multi-objective optimization problems. However, in high-density traffic scenarios at ramps, constructing accurate nonlinear MPC models poses enormous online computational challenges. Complex nonlinear dynamic constraints and time-varying collision avoidance constraints result in time-consuming numerical solution processes, which are difficult to meet (near-)real-time control demands. Additionally, existing distributed cooperative methods often assume vehicles travel along fixed paths, ignoring the flexible path planning requirements during ramp merging (e.g., dynamic lane changing, smooth path switching) [2–4]. This lack of flexibility leaves vehicles unable to effectively respond to sudden conflicts.

To tackle the aforementioned challenges, this study proposes a distributed multi-vehicle cooperative trajectory planning and control framework for ramp merging scenarios, integrating Deep Neural Networks (DNNs) with MPC. The main contributions of this paper are as follows:

1. The primary technical contribution is the real-time feasibility achieved by the neural approximation. While maintaining tracking accuracy comparable to numerical MPC, the proposed method reduces computation time by orders of magnitude, enabling deployment in dynamic, fast-paced ramp merging.
2. A distributed cooperative framework based on dynamic topology is constructed to effectively reduce communication loads.
3. A feature point-based Cubic Bézier Curve trajectory generation method is proposed, which not only ensures path smoothness but also avoids over-reliance on global high-precision maps, enabling flexible path planning in complex ramp environments.

2. Related Work

Vehicle cooperative control in ramp merging scenarios is an interdisciplinary field involving traffic engineering, control theory, and artificial intelligence. This section reviews relevant research from three aspects: vehicle operating characteristics in ramp merging scenarios, multi-vehicle cooperative control architectures, and learning-based optimal control methods.

2.1. Vehicle Operating Characteristics in Ramp Merging Scenarios

Due to their special geometric structures and complex traffic flow characteristics, ramp merging areas have long been a research focus in traffic control. A deep understanding of the microscopic behavior of vehicles in these areas is a prerequisite for designing cooperative strategies. Daamen et al. [9] found through empirical analysis that most drivers tend to complete lane-changing maneuvers in the initial segment of the acceleration lane, and the distribution of lane-changing positions is significantly influenced by main-line traffic flow. Further studies by Calvi et al. [10] and Marczak et al. [11] indicated that the length of the acceleration lane has a limited impact on driving behavior. Under smooth traffic conditions, approximately half of the acceleration lane length is underutilized, suggesting room for optimizing the utilization of existing infrastructure.

In terms of driving interactions, The authors of references [12,13] pointed out that significant differences exist in lane-changing strategies among different drivers, and this heterogeneity increases the difficulty of prediction and control. Wan et al. [14] emphasized that when merging under constrained conditions, vehicles need to adjust not only their speed but also the gap with preceding vehicles dynamically (Gap Acceptance). Lyu Jiangbi [15] comprehensively evaluated the safety of ramp merging areas, considering factors such as driver behavior and traffic conflicts. G.F. [16] constructed a ramp merging model through real-vehicle experiments and found that up to 35% of vehicles perform consecutive illegal lane changes after merging into the mainline, imposing higher compliance requirements on trajectory planning in cooperative control.

2.2. Cooperative Control Architectures for Connected Multi-Vehicle Systems

Existing multi-vehicle cooperative control architectures are primarily divided into centralized and distributed types.

The core of centralized control lies in collecting all vehicle information through a central node and computing the globally optimal strategy. Dresner [17] proposed an early reservation-based conflict management scheme, which divides conflict areas into grids and coordinates the passing order through vehicle requests. Marinescu [18] applied this reservation method to ramp merging environments, achieving a reliable cooperative mechanism. However, Rios-Torres [19] and Vasirani [20] noted that centralized methods face challenges such as high communication loads, low fault tolerance, and potential

deadlock issues. Particularly in large-scale complex scenarios, the central node bears enormous computational pressure. To address optimization difficulties, The authors of references [21–25] proposed modeling the coupled constraints of vehicles in the road network using the Alternating Direction Method of Multipliers (ADMM), attempting to reduce the complexity of centralized computing.

Distributed architectures have gradually become mainstream due to their robustness and flexibility. Maestre [26] pointed out that distributed frameworks outperform centralized schemes in scalability and fault tolerance. In early research on vehicle platoon control, Lee [21] and Zheng [25] summarized and proposed the classic four-element architecture (information flow topology, distributed controller, node dynamics, and geometric configuration), laying a theoretical foundation for distributed control.

For complex scenarios such as ramps and intersections, The authors of references [27–29] proposed the concept of “Virtual Vehicle,” realizing conflict resolution by mapping virtual nodes. Wang [30] and Zhang [31] further extended the virtual platoon method, optimizing traffic flow using the First-Come-First-Served (FCFS) strategy and cooperative lane-changing logic, respectively. In terms of Distributed Model Predictive Control (DMPC), the authors of references [32–35] proposed a priority-based non-cooperative DMPC method, emphasizing safety without explicit cooperation mechanisms. GUO et al. [36] proposed a distributed observation and trajectory tracking strategy for unsignalized intersections, while Xu [37] introduced a depth-limited search algorithm to construct conflict-free spanning trees. However, most of the aforementioned distributed methods [38–42] assume vehicles travel along fixed paths, lacking the ability to flexibly handle the coupling of lateral lane changing and longitudinal acceleration in ramp scenarios.

2.3. Learning-Based Control Strategy Solving and Optimization

Although MPC can handle multi-constraint problems, its real-time performance remains the biggest bottleneck for practical deployment. To accelerate the solving process, the authors of references [43–46] attempted online acceleration techniques such as warm-start and early termination of interior-point methods, but these methods are still limited by the quality of the initial solution.

In recent years, Approximate Dynamic Programming (ADP) and neural network approximation methods have provided new ideas for addressing this issue. Lin [47] and Duan [48] designed ADP algorithms with finite time horizons and state constraints to solve nonlinear control problems, respectively. Lucia et al. [49,50] proposed an imitation learning strategy based on explicit dynamic models, using neural networks to approximate MPC laws. Zhang et al. [51] adopted supervised learning to approximate MPC strategies for linear parameter-varying (LPV) systems, while Liu et al. [52] utilized recurrent neural networks (RNNs) to handle finite-horizon tracking tasks. The authors of references [53–55] further integrated reinforcement learning with generalized exterior point penalty functions to handle collision constraints. These studies demonstrate that using deep neural networks to approximate the MPC optimization process (NN-MPC) not only retains MPC’s ability to handle constraints but also reduces computation time to the millisecond level. This is an effective approach to solving complex nonlinear control problems in ramp merging scenarios. Based on this insight, this paper constructs a cooperative control framework integrating deep neural networks with distributed MPC.

3. Methodology

3.1. Framework Overview

This study aims to address the issues of low efficiency and insufficient safety in multi-vehicle cooperative driving within the ramp merging scenario. To this end, a multi-vehicle

cooperative trajectory planning and control framework based on Distributed Model Predictive Control (DMPC) is proposed. As illustrated in Figure 1, the framework comprises three core modules: the environment perception and inter-vehicle communication module, the cooperative trajectory generation module, and the distributed motion control module.

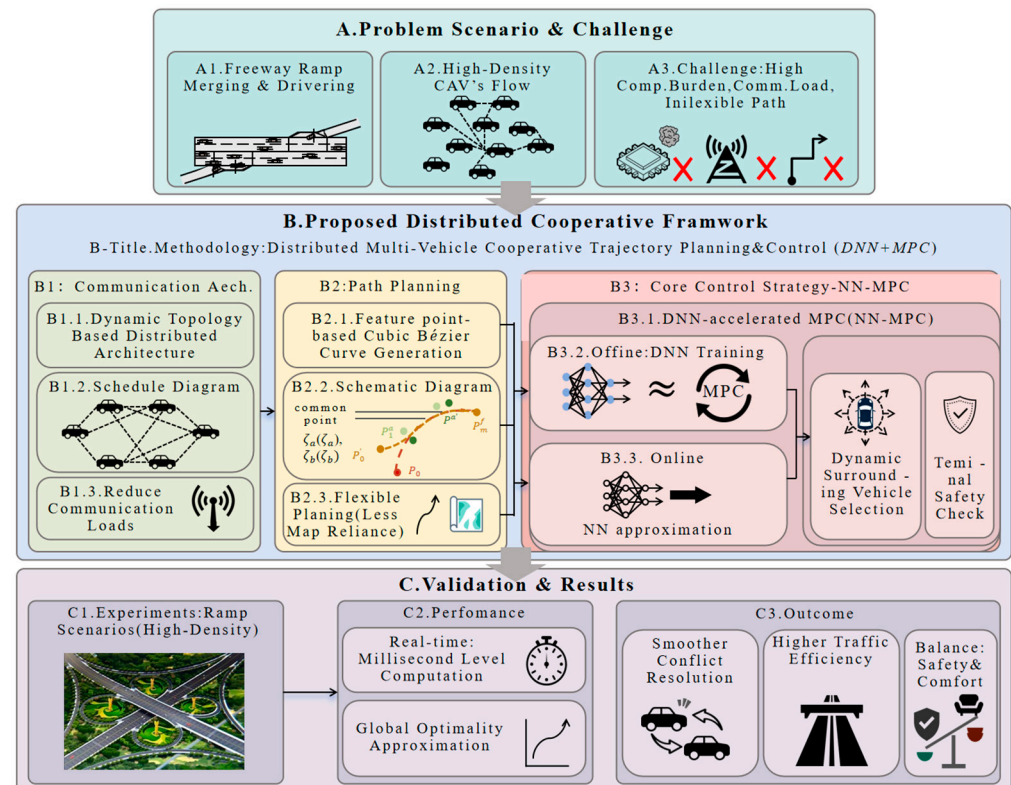


Figure 1. Framework of the proposed method.

The environment perception and inter-vehicle communication module is tasked with constructing the local dynamic traffic topology. It ensures that vehicles within the communication range share their planned trajectory information in real time, rather than merely their current instantaneous states, thereby providing deterministic intent inputs for downstream decision-making. The cooperative trajectory generation module utilizes static road network constraints and dynamic interaction information, employing Cubic Bézier Curves based on a warm-start strategy to generate spatiotemporally consistent reference trajectories. Under the premise of satisfying vehicle dynamics and collision avoidance constraints, the distributed motion control module solves for optimal control inputs through receding horizon optimization, achieving accurate tracking of reference trajectories and conflict resolution.

Assumption 1 (Full Connectivity). To ensure the focus remains on the algorithmic performance, a key operational assumption is made: To isolate the efficacy of the proposed cooperative mechanism, this study assumes a 100% penetration rate of Connected and Automated Vehicles (CAVs). The interaction with human-driven vehicles (mixed traffic) introduces prediction uncertainties that are outside the scope of this specific work and will be addressed in future studies.

3.2. Environment Perception and Interaction Topology Construction

Given the requirements for real-time performance and integrity of information interaction in multi-vehicle cooperative systems, this study first establishes a platoon interaction mechanism based on communication topology. It further integrates local environmental feature points to achieve the digital reconstruction of the ramp merging scenario.

3.2.1. Information Flow Topology Construction

The connected and automated vehicle platoon in this study is modeled as a self-organizing network. Following the study by references [56–58], the communication range is set to 300 m, and vehicles within this range with a relative speed of no more than 200 km/h can establish stable data connections. (Note: The effective Look-ahead Distance is governed by the V2X communication range, set to 300 m. Vehicles within this range, even those approaching at high speeds, are captured by the dynamic topology graph $G(t)$.) Unlike traditional methods, this study takes the planned trajectory sequence of vehicles within the prediction horizon N_p as the core communication content. Each frame of communication data includes the ego-vehicle state vector and, crucially, the predicted trajectory sequence over the entire prediction horizon N_p (i.e., $\mathbf{x}_{t:t+N_p}$), rather than just the instantaneous state. This enables neighboring vehicles to anticipate future conflicts. The communication topology is denoted as:

$$G(t) = (V, E(t), A(t)) \quad (1)$$

where the node set $V = \{V_1, V_2, V_3, \dots, V_N\}$ treats each vehicle as an individual node; the time-varying edge set $E(t) \subseteq V \times V$, with an edge $(i, j) \in E(t)$ indicating a communication connection between node i and node j ; and the adjacency matrix $A(t) = [a_{ij}(t)] \in R^{N \times N}$ quantifies the connection relationship between nodes: $a_{ij}(t)$ if $(i, j) \in E(t)$ (i.e., vehicle i and vehicle j communicate with each other), and $a_{ij}(t) = 0$ otherwise (no communication between the two vehicles). Notably, this study adopts a neighborhood communication mechanism to construct the communication topology, which ensures the adjacency matrix $A(t)$ is always symmetric, i.e., $A(t) \equiv A^T(t)$.

Under this topological structure, the “neighborhood set” of vehicle i at any time t is defined as the subset of vehicles that can directly exchange information with it:

$$N_i(t) := \{j \mid a_{ij}(t) = 1\} \quad (2)$$

This mechanism ensures that each vehicle only requires processing local interaction information within its neighborhood set. While effectively managing the communication load of large-scale platoons, it maintains the stability of the computational complexity of the distributed algorithm, thereby enhancing the system’s adaptability to dynamic traffic conditions. Although ideal communication is assumed for training, the Receding Horizon nature of the proposed framework inherently mitigates the impact of transient packet loss by executing the valid portion of the previously predicted trajectory sequence.

3.2.2. Modeling of Local Environmental Features in Ramp Merging Scenarios

To standardize the operating environment of ramp merging scenarios, this study divides the merging area into four functional zones (as shown in Figure 2): the Normal Driving Zone, the Preparatory Control Zone, the Key Control Zone, and the Vehicle Merging Zone. For high-precision trajectory planning, vehicles need to acquire sparse feature points of the lane centerlines of lane centerlines and boundaries in real time. As illustrated in Figure 3, by collecting coordinates of feature points in the Key Control Zone and generating a basic reference path using smooth connecting lines, a reference trajectory sequence containing geometric and speed information is constructed by integrating static

constraints such as section speed limits and traffic light phases. This environmental modeling process fully considers road geometric constraints, providing an accurate input benchmark for subsequent lane-changing, overtaking, and collision avoidance decisions.

Assumption 2 (Constant Friction). A constant tire–road friction coefficient (approx. $\mu = 0.8$ for dry asphalt) is assumed. While varying surface conditions (e.g., wet roads) affect low-level tracking, the proposed MPC framework operates at the trajectory planning layer, where robust margins are embedded to tolerate minor friction variations.

Assumption 3 (Ideal Actuation). Mechanical braking delays (typically ~ 200 ms) are not explicitly modeled to isolate algorithmic performance. However, the Receding Horizon Control (RHC) mechanism inherently mitigates the impact of such delays by constantly re-planning based on the current state feedback.

Assumption 4 (Kinematic Model Validity). The Nonlinear Dynamic Bicycle Model is employed for trajectory planning. Since the ramp merging maneuvers involve smooth steering and moderate speeds (without extreme lateral accelerations that induce significant tire slip angles), this kinematic approximation provides a sufficient balance between prediction accuracy and computational simplicity.

Assumption 5 (Sensing and Noise). We assume high-precision positioning (e.g., RTK-GPS) is available. While real-world sensors have noise, the feedback loop of MPC is known to reject zero-mean Gaussian noise effectively. Therefore, synthetic noise was not explicitly added to the simulation inputs to maintain the clarity of the algorithmic evaluation and isolate the controller's performance.

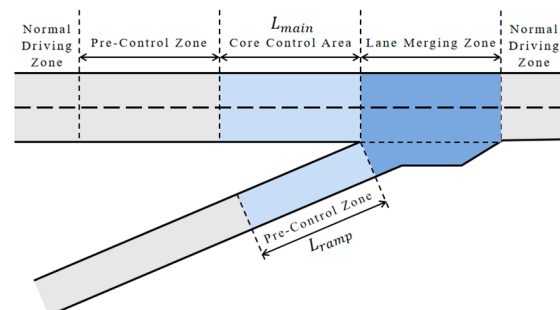


Figure 2. Control Zones of the Ramp Merging Scenario.

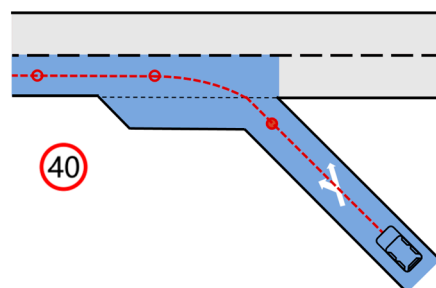


Figure 3. Element Framework of the Connected Vehicle Platoon Under Geometric Constraints of the Road Network.

3.3. Scenario Reference Trajectory Generation

Prior to elaborating on the specific path generation algorithm, it is essential to clarify the underlying coordination mechanism of the proposed framework. This distributed coordination relies on an implicit negotiation mechanism learned by the policy network. Unlike explicit priority-based rules, the NN-MPC agent assesses the intent of surrounding vehicles by utilizing shared predicted trajectories. The asymmetry in the state space (e.g., differences in the distance to the merge point) naturally guides the agents to converge towards distinct roles (yielding vs. passing), effectively preventing deadlocks and conflicting local plans without the need for a central coordinator. Based on this coordination logic, the reference trajectory for each vehicle is generated by the system to execute these high-level intents.

3.3.1. Path Generation

To address the reliance on environmental information, it is important to first clarify the map data requirements of this framework. The proposed method does not rely on dense point-cloud high-precision maps typically used for heavy localization. Instead, it utilizes a lightweight vector map containing only lane centerlines and topology connectivity, which is assumed to be available offline. Real-time localization is achieved via on-board sensors relative to these vector features, making the system robust to GPS drift.

Based on this vector topology, to ensure the smoothness and continuity of the planned path, this study adopts Cubic Bézier Curves to describe the vehicle's local path. Let the pose of the vehicle at the current time t be $P_0 = [X_0, Y_0, \varphi_0]^T$. Since planning is always performed at the current time and outputs the reference trajectory tracked by the prediction model within the future prediction horizon, the time t is omitted here to simplify the notation. As illustrated in Figure 4, the vehicle can simultaneously obtain the path feature point $P_m^f = [X_m^f, Y_m^f, \varphi_m^f]^T$, where f denotes the index of the feature point. As the vehicle travels into the neighborhood of the current path point P_m^f , the index m is updated to $(m + 1)$, switching to the next feature point. Feature points can either be stored in the offline map or identified online by integrating the global navigation path and the perception-localization system. The local path $\mathcal{R}(\zeta)$ is parameterized by the curve parameter ζ and determined by four control points: the starting point P_0 , two intermediate auxiliary points P_1^a and P_2^a , and the target feature point P_m^f . The trajectory is defined piecewise as follows:

for $\zeta \in [0, 1]$:

$$\mathcal{R}(\zeta \mid P_0, P_1^a, P_2^a, P_m^f) = \begin{bmatrix} \mathcal{R}_X(\zeta \mid P_0, P_1^a, P_2^a, P_m^f) \\ \mathcal{R}_Y(\zeta \mid P_0, P_1^a, P_2^a, P_m^f) \end{bmatrix} = \begin{bmatrix} X_0, 3X_1^a, 3X_2^a, X_m^f \\ Y_0, 3Y_1^a, 3Y_2^a, Y_m^f \end{bmatrix} \begin{bmatrix} (1-\zeta)^3 \\ (1-\zeta)^2\zeta \\ (1-\zeta)\zeta^2 \\ \zeta^3 \end{bmatrix} \quad (3)$$

for $\zeta \in [1, +\infty]$:

$$\mathcal{R}(\zeta \mid P_m^f) = \begin{bmatrix} \mathcal{R}_X(\zeta \mid P_m^f) \\ \mathcal{R}_Y(\zeta \mid P_m^f) \end{bmatrix} = \begin{bmatrix} X_m^f \cos \varphi_m^f \\ Y_m^f \sin \varphi_m^f \end{bmatrix} \begin{bmatrix} 1 \\ \zeta^{-1} \end{bmatrix} \quad (4)$$

Specifically, to satisfy the geometric constraints where the reference path must be tangent to the vehicle's heading at P_0 and the lane direction at P_m^f , the coordinates of the auxiliary points are determined by:

$$\begin{cases} X_1^a = X_0 + L_0 \cos \varphi_0 \\ Y_1^a = Y_0 + L_0 \sin \varphi_0 \end{cases} \quad \begin{cases} X_2^a = X_m^f - L_3 \cos \varphi_m^f \\ Y_2^a = Y_m^f - L_3 \sin \varphi_m^f \end{cases} \quad (5)$$

where L_0 and L_3 represent the tangent lengths at the start and end points, respectively.

This geometric constraint, combined with the inherent smoothness of the Cubic Bézier Curves, ensures that the generated reference path satisfies the basic curvature and

boundary requirements before being tracked by the controller. Thus, the feasibility of the reference path is guaranteed by design.

It is noteworthy that the Cubic Bézier generation serves as coarse-grained guidance. While it ensures geometric smoothness, the fine-grained collision avoidance is explicitly handled by the downstream MPC constraints. The ‘Historical Trajectory Reuse’ strategy minimizes the conflict between the reference path and the obstacle-avoiding trajectory by dynamically updating the start point P'_0 based on the vehicle’s actual state.

Remark on Control Continuity. *It is important to note that the proposed framework employs a unified controller rather than switching between separate ramp and mainline controllers. The global reference trajectory (generated via Bézier curves) provides a seamless path from the ramp to the mainline, ensuring continuous control without discrete handoff coordinates.*

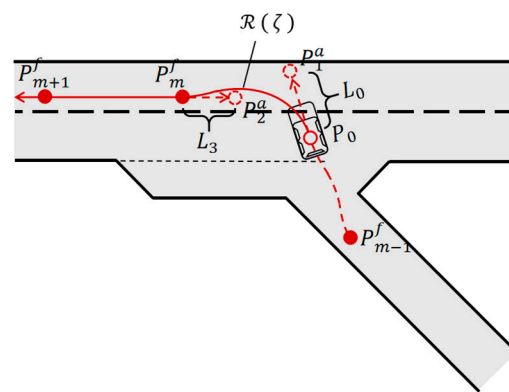


Figure 4. Local Path Planning Based on Road Feature Points.

To ensure that the newly generated reference path is consistent with the vehicle’s historical driving intent, this study proposes an iterative planning algorithm based on historical trajectory reuse. The core idea is that the first half of the new path planned at the current time step t should be as close as possible to the remaining part of the path planned at the previous time step $(t - 1)$.

Assume that the path function generated at the previous time step is R_{prev} and the path to be solved at the current time step is R_{new} . Based on the properties of Cubic Bézier Curves, the auxiliary points P_1^a and P_2^a can be solved to ensure that the two curves coincide at specific parameter points:

$$\mathcal{R}_{prev}(\zeta|P_0, P_{1a}, P_{2a}, P_m^f) = \mathcal{R}_{new}(\zeta'|P_0', P_{1a}', P_{2a}', P_m^f) \quad (6)$$

Among these, P_0 , P_1^a , P_2^a and P_m^f are known. The above equation characterizes the condition under which the trajectory planned at the previous step (parameterized by ζ coincides with the newly generated trajectory, parameterized by ζ' at a specific shared point).

As shown in Figure 5, due to control deviations or obstacle avoidance, the updated vehicle position P_0' may not strictly fall on the previous reference path. To address the potential deviation shown in Figure 5, the evaluation parameters ζ_a and ζ_b are selected within the interval $[0.5, 1)$. Selecting larger values in this interval reduces the reliance on the path segment near the vehicle’s current position—which is susceptible to disturbances—and places greater weight on the future trajectory trend. This strategy ensures that the algorithm maintains path continuity while exhibiting robustness against control execution errors.

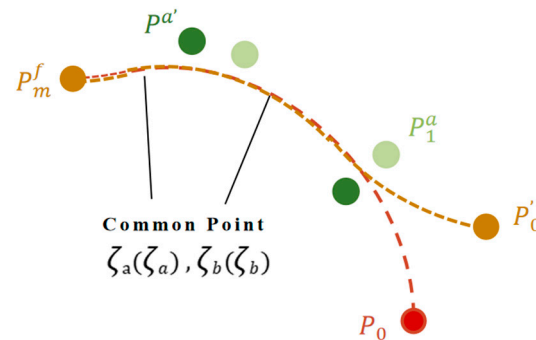


Figure 5. Schematic Diagram of Iterative Planning for Cubic Bézier Curves.

To solve for $P_1^{a'}$ and $P_2^{a'}$, two evaluation points ζ_a and ζ_b are selected on the new path. As the vehicle travels forward, there exists a mapping relationship between the new path parameter ζ' and the old path parameter ζ . Based on the assumption of arc length preservation, this mapping relationship can be approximately estimated as:

$$\frac{1 - \zeta'}{1 - \zeta} = \frac{d_{P_m^f P_0}}{d_{P_m^f P_0'}} \approx \frac{1}{1 - \zeta_{first}} \quad (7)$$

Herein, $d_{P_m^f P_0}$ denotes the arc length from point P_m^f to P_0 , and $d_{P_m^f P_0'}$ denotes the arc length from point P_m^f to P_0' . The ratio of the arc lengths can be approximated by the ratio of the parameters, where ζ_{first} is the ζ value corresponding to the first point of the planned trajectory at the current time step.

By combining the above conditions, the geometric constraints can be transformed into a linear solution problem regarding the coordinate vectors of the auxiliary points. Let $L_0 = \|P_0' - P_1^{a'}\|_2$ and $L_3 = \|P_2^{a'} - P_m^f\|_2$ represent the tangent lengths of the control points, respectively. Based on the derivative properties of Cubic Bézier Curves, the optimal L_0 and L_3 can be quickly solved via matrix inversion:

$$\begin{bmatrix} L_0 \\ L_3 \end{bmatrix} = A_{X,Y}^{-1} b_{X,Y} \quad (8)$$

The definitions of each component are as follows: the “0” in matrix $A_{X,Y}$ denotes a 2×2 zero matrix.

$$A_{X,Y} = \begin{bmatrix} A(\zeta_a, \zeta_b) & 0 \\ 0 & A(\zeta_a, \zeta_b) \end{bmatrix} \quad (9)$$

$$A(\zeta_a, \zeta_b) = \begin{bmatrix} 3\zeta_a(1-\zeta_a)^2 & 3\zeta_a^2(1-\zeta_a) \\ 3\zeta_b(1-\zeta_b)^2 & 3\zeta_b^2(1-\zeta_b) \end{bmatrix} \quad (10)$$

$$b_{X,Y} = \begin{bmatrix} R_X(\zeta_a / P_0, P_1^a, P_2^a, P_m^f) - X_0'(1-\zeta_a)^3 - X_m^f \zeta_a^3 \\ R_X(\zeta_b / P_0, P_1^a, P_2^a, P_m^f) - X_0'(1-\zeta_b)^3 - X_m^f \zeta_b^3 \\ R_Y(\zeta_a / P_0, P_1^a, P_2^a, P_m^f) - Y_0'(1-\zeta_a)^3 - Y_m^f \zeta_a^3 \\ R_Y(\zeta_b / P_0, P_1^a, P_2^a, P_m^f) - Y_0'(1-\zeta_b)^3 - Y_m^f \zeta_b^3 \end{bmatrix} \quad (11)$$

Once the continuous path curve is determined, the reference trajectory sequence is discretized by performing piecewise integration of the speed profile function within the prediction horizon.

3.3.2. Dynamic Speed Planning

After the geometric path is determined, it is necessary to endow the path with temporal attributes based on road speed limits and traffic flow states. To guide the longitudinal motion in specific merging areas, Table 1 defines the velocity planning rules for vehicles under different ramp scenarios. Based on these rules, Table 2 presents the desired speed function $U^{ref}(t)$ of the vehicle at time t , which is determined by the remaining phase time T_{left} and the current section speed limit U_{limit} . Here, s denotes the distance from the vehicle to the intersection stop line at the current time step: $s < 0$ indicates that the vehicle has crossed the ramp exit stop line and entered the intersection; U is the observed value of the vehicle's longitudinal speed at the current time step; when the vehicle exits the intersection, s is updated to the distance from the connected and automated vehicle (CAV) to the stop line of the next intersection; L_{bound} is the intersection distance threshold, satisfying $L_{bound} \leq T_{phase} U_{limit}$; T_{phase} denotes the duration of the single phase of the traffic light; a_{max} is the maximum longitudinal acceleration of the CAV.

After obtaining the continuous speed profile function $U^{ref}(t)$ from Table 2, the speed sequence can be derived as:

$$U_{seq} = [U^{ref}(0), \dots, U^{ref}(kT_s), \dots, U^{ref}((N_p - 1)T_s)] \quad (12)$$

Table 1. Vehicle's Uniformly Variable Speed Planning Rules in Ramp Merging Scenarios.

Ramp Scenario			Velocity Function $U^{\text{ref}}(t)$
$L_{\text{bound}} \leq s$	No Straight-Moving Vehicles After T_{left}	$UT_{\text{left}} > s$	$0.9 U_{\text{limit}}$
		$UT_{\text{left}} \leq s$	$\max \left\{ U - \frac{U^2}{2s} t, 0 \right\}$
	Straight-Moving Vehicles Exist at T_{left}	$U_{\text{limit}} T_{\text{left}} > s$	$\max \left\{ \frac{s}{T_{\text{left}}}, U_{\text{limit}} \right\}$
		$U_{\text{limit}} T_{\text{left}} \leq s$	$\max \left\{ U - \frac{U^2}{2s} t, 0 \right\}$
$s < 0$			$0.9 U_{\text{limit}}$

Table 2. Comparison of Mainstream Cooperative Control Approaches.

Approach	Mechanics	Pros	Cons
Rule-based	Pre-defined logic (e.g., FCFS)	Simple, low computation	Poor flexibility, prone to deadlocks in high density
Standard MPC	Online numerical optimization	Handles constraints, optimal tracking	Computationally heavy, difficult for real-time deployment
Pure RL	Trial-and-error learning	Real-time inference	"Black-box" safety risks, lack of constraint guarantees
Proposed NN-MPC	Offline Optimization and Online Inference	Real-time (<10 ms), Explicit Safety Constraints	Dependence on training data distribution

3.3.3. Trajectory Generation

Algorithm 1 presents two iterative reference trajectory generation methods. The first is a generation method based on a small number of road feature points. This method integrates path planning and speed planning, taking key road feature points on the global

path as inputs. When the vehicle approaches these feature points, the reference trajectory adjusts accordingly and converges to the feature points. Even if the vehicle deviates from the preset path within the permissible driving area, it can return to the path endpoint after exiting the conflict zone. The second is a generation method based on the sequence of lane center points. The core idea of this method is to define circular arcs intersecting the lane centerline, and select the discrete point of the lane centerline closest to the intersection point as the endpoint of the planned Cubic Bézier Curves.

Algorithm 1 achieves real-time trajectory planning based on road characteristics. The key to the algorithm lies in the optimization of the path function and the update of feature points, ensuring that vehicles can drive safely and smoothly in the dynamic ramp environment.

Algorithm 1: Path Function and Feature Point Update Process

Input: Prediction horizon N_p ; current pos P_0 ; feature point P_m^f ; speed sequence U_{seq} ; time step T_s ; intermediate variables from previous iteration L_0 and L_3 ; constant $L > 0$, $\zeta_a, \zeta_b \in [0.5, 1)$ with $\zeta_a \neq \zeta_b$; update threshold $\zeta_{bound} \in (0, 1)$; update flag $UpdateFeaturePoint$

Output: New path function $R_{new}(\zeta)$, new intermediate variables L_0^{new} and L_3^{new} , update feature point flag $UpdateFeaturePoint$

```

1:   $UpdateFeaturePoint \leftarrow False$ 
2:  Initialize the  $\zeta := [\zeta_0, \zeta_1, \dots, \zeta_{N_p}] \leftarrow [0, 0, \dots, 0]$ 
3:  Calculate  $P_1^a$  and  $P_2^a$ 
4:  Construct path function  $R_{new}(\cdot) \leftarrow R(\cdot | P_0, P_1^a, P_2^a, P_m^f)$ 
5:  for  $idx=1$  to  $N_p$  do
6:      Calculate speed change  $\Delta L = U_{seq}[idx-1] \cdot T_s$ 
7:      Determine  $\zeta_{idx}$  using binary search such that:
          
$$\left| \|R_{new}(\zeta_{idx}), R_{new}(\zeta_{idx-1})\|_2 - \Delta L \right| \leq ErrorRate \cdot \Delta L$$

8:      end for
9:      if  $\zeta_1 > \zeta_{bound}$  then
10:          $UpdateFeaturePoint \leftarrow True$ 
11:         Update intermediate variables:  $L_0^{new} \leftarrow L$ ,  $L_3^{new} \leftarrow L$ 
12:      else
13:         Solve the linear system of equations to update parameters:
            
$$\zeta'_a \leftarrow (\zeta_1 - \zeta_a) / (\zeta_1 - 1), \quad \zeta'_b \leftarrow (\zeta_1 - \zeta_b) / (\zeta_1 - 1)$$

14:         Obtain solutions of the linear system (5):  $X_1^a, X_2^a, Y_1^a, Y_2^a$ 
15:         Calculate new intermediate variables  $L_0^{new}$  and  $L_3^{new}$ 
16:      end if
17:  return  $R_{new}(\zeta)$ ,  $L_0^{new}$ ,  $L_3^{new}$ ,  $UpdateFeaturePoint$ 
```

3.4. Distributed Model Predictive Control (DMPC) Design

Considering that traditional multi-vehicle cooperative control methods struggle to adapt to the dynamic changes in vehicle states, fail to effectively handle complex constraints, and lack sufficient robustness against environmental changes and execution errors, while also facing difficulties in resolving potential multi-vehicle conflicts through explicit driving intent, this study proposes a hierarchical and progressive distributed control framework, the overall architecture of which is illustrated in Figure 6. First, an optimization problem incorporating vehicle dynamics and safety collision avoidance constraints is formulated based on Model Predictive Control (MPC) to ensure the theoretical safety and rationality of cooperative driving behaviors. Subsequently, a deep neural net-

work (DNN) policy approximation method is introduced to address the low computational efficiency of traditional numerical solvers in complex scenarios. Finally, a dynamic surrounding vehicle selection mechanism is designed to enhance the system's adaptability and flexibility to changes in traffic flow topology.

To ensure high-fidelity tracking accuracy and realistic dynamic response, a nonlinear dynamic bicycle model is employed as the predictive model for the MPC controller. Unlike kinematic models, this model explicitly accounts for tire slip angles and lateral dynamics, which are critical in ramp merging maneuvers.

3.4.1. Modeling of the Cooperative Control Problem

Based on the system architecture, the vehicle state vector at time step k is defined as $\mathbf{x}_k = [X_k, Y_k, \varphi_k, U_k, V_k, \omega_k]^T$, where (X_k, Y_k) denotes the global position, φ_k is the yaw angle, U_k and V_k are the longitudinal and lateral velocities in the vehicle body frame, and ω_k is the yaw rate. The control input vector is defined as $\mathbf{u}_k = [a_k, \delta_k]^T$, representing the longitudinal acceleration and the front wheel steering angle, respectively.

The continuous-time nonlinear dynamics are governed by the following differential equations, utilizing the parameters defined in Table 5:

$$\begin{aligned}\dot{X} &= U \cos \varphi - V \sin \varphi \\ \dot{Y} &= U \sin \varphi + V \cos \varphi \\ \dot{\varphi} &= \omega \\ \dot{U} &= a - \frac{F_{yf} \sin \delta}{m} + V \omega \\ \dot{V} &= \frac{F_{yf} \cos \delta + F_{yr}}{m} - U \omega \\ \dot{\omega} &= \frac{l_f F_{yf} \cos \delta - l_r F_{yr}}{I_z}\end{aligned}\quad (13)$$

The lateral tire forces F_{yf} and F_{yr} are approximated using a linear tire model, which is valid for the operational range of ramp merging:

$$\begin{aligned}F_{yf} &= 2k_f \left(\delta - \arctan \left(\frac{V + l_f \omega}{U} \right) \right) \\ F_{yr} &= 2k_r \left(-\arctan \left(\frac{V - l_r \omega}{U} \right) \right)\end{aligned}\quad (14)$$

where m is the vehicle mass, I_z is the yaw moment of inertia, l_f and l_r are the distances from the center of mass to the front and rear axles, and k_f and k_r are the equivalent cornering stiffnesses of the front and rear tires, as listed in Table 5.

This section establishes a receding horizon optimization method where each vehicle in the multi-vehicle system is independently equipped with an MPC controller. Only the first optimal control action is executed in each control cycle, and the optimal policy is recalculated in the next cycle based on the latest ego-vehicle state, surrounding vehicle states, and the updated reference trajectory. To ensure real-time performance, the coordination adopts a non-iterative (single-shot) information exchange protocol per time step to minimize communication latency. Although this introduces minor prediction mismatches, the Receding Horizon Control (RHC) mechanism inherently compensates for these errors by re-optimizing trajectories at the subsequent step.

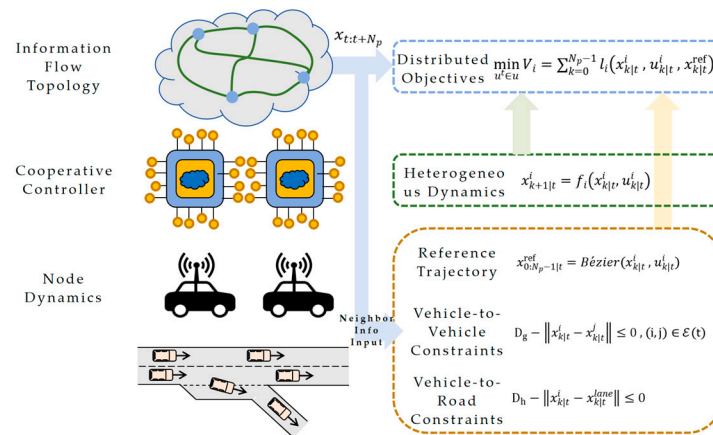


Figure 6. Distributed Model Predictive Control (DMPC) Scheme.

In this study, the number of planned reference trajectories for each vehicle is set to 1. Based on the architecture of the integrated decision-control scheme and combined with the problem formulation of cooperative control, the objective function of cooperative controller i is given as follows:

$$\min_{u^i} V(x^i, x^{ref}) = \sum_{k=0}^{N_p-1} \ell_{track}(x_{k|t}^i, x_{k|t}^{ref}) + \ell_{energy}(u_{k|t}^i) \quad (15)$$

where $u_{k|t}^i$ and $x_{k|t}^i$ denote the control and state variables of vehicle i at the k -th step within the prediction horizon, respectively, and $x_{k|t}^{ref}$ represents the reference trajectory planned for vehicle i at time t . Here, ℓ_{track} denotes the tracking accuracy, ℓ_{energy} denotes the fuel consumption, and $\ell_{comfort}$ denotes performance indicators such as driving comfort. To prevent erratic vehicle behavior and ensure smooth control despite potential sub-optimality in the neural approximation, the cost function explicitly penalizes high-frequency control fluctuations. Specifically, the weight matrix R (associated with $\ell_{comfort}$) imposes heavy penalties on the rate of change of control inputs (Δu), thereby physically constraining the DNN to output smooth and continuous trajectories.

To achieve the desired cooperative behavior, the MPC optimization problem is formulated with a multi-objective cost function that explicitly balances tracking accuracy, energy efficiency, driving comfort, and safety. The full objective function J over the prediction horizon N_p is defined as follows:

$$J(\mathbf{x}(t), \mathbf{u}(\cdot)) = \sum_{k=0}^{N_p-1} \left(\underbrace{\|\mathbf{x}_{k|t} - \mathbf{x}_{k|t}^{ref}\|_{\mathbf{Q}}^2}_{\text{Tracking}} + \underbrace{\|\mathbf{u}_{k|t}\|_{\mathbf{R}}^2}_{\text{Energy}} + \underbrace{\|\Delta \mathbf{u}_{k|t}\|_{\mathbf{S}}^2}_{\text{Comfort}} + \underbrace{\rho \cdot P_{coll}(\mathbf{x}_{k|t})}_{\text{Safety}} \right) + \underbrace{\|\mathbf{x}_{N_p|t} - \mathbf{x}_{N_p|t}^{ref}\|_{\mathbf{Q}_f}^2}_{\text{Terminal}} \quad (16)$$

where $\mathbf{x}_{k|t}$ and $\mathbf{x}_{k|t}^{ref}$ are the predicted and reference state vectors at step k , respectively. $\mathbf{u}_{k|t}$ is the control input vector, and $\Delta \mathbf{u}_{k|t} = \mathbf{u}_{k|t} - \mathbf{u}_{k-1|t}$ represents the rate of change of control inputs, which serves as the smoothness term to ensure passenger comfort. $P_{coll}(\cdot)$ is a soft penalty function for collision avoidance constraints (as defined in Equation (16)) with a high penalty weight ρ .

The weighting matrices are defined as diagonal matrices to normalize and prioritize different objectives: $\mathbf{Q} = \text{diag}(w_x, w_y, w_\phi, w_v, \dots)$, $\mathbf{R} = \text{diag}(w_{acc}, w_\delta)$, and $\mathbf{S} = \text{diag}(w_{\Delta acc}, w_{\Delta \delta})$. The specific numerical values for all weight terms used in the simulation are reported in Table 3.

Table 3. Numerical Weight Values for the MPC Objective Function.

Cost Term	Parameter	Symbol	Value	Physical Meaning
Tracking	Position Weight	w_{pos}	10.0	Penalizes lateral/longitudinal deviation ($1/m^2$)
	Heading Weight	w_{φ}	50.0	Penalizes yaw angle error ($1/rad^2$)
	Velocity Weight	w_v	1.0	Penalizes speed tracking error ($1/(m/s)^2$)
Energy	Acceleration Weight	w_{acc}	0.1	Penalizes magnitude of acceleration
	Steering Weight	w_{δ}	0.1	Penalizes magnitude of steering angle
Comfort	Jerk Weight	$w_{\Delta acc}$	100.0	Penalizes rate of acceleration change (Smoothness)
	Steering Rate Weight	$w_{\Delta \delta}$	500.0	Penalizes rate of steering change (Smoothness)
Safety	Collision Penalty	ρ	10,000.0	Soft constraint penalty strength for collision risk
Terminal	Terminal Weight	Q_f	10.0	Terminal state cost (typically scaled Q)

To ensure control effectiveness and safety, four types of constraint conditions are established.

Equality and inequality constraints are formulated based on vehicle dynamics, traffic legality in ramp merging scenarios, and driving safety on ramps. The vehicle dynamics model f_i serves as the equality constraint relating the control vector $u_{k/t}^i$ and the state vector $x_{k/t}^i$:

$$x_{k+1/t}^i = f_i(x_{k/t}^i, u_{k/t}^i) \quad (17)$$

For traffic regulation constraints, let $h(\cdot, \cdot)$ denote a scalar function of the vehicle state vector $x_{k/t}^i$ and the local road feature vector:

$$h(x_{k/t}^i, x_{k/t}^{road, i}) \leq 0 \quad (18)$$

In the research on multi-vehicle cooperative driving and collision avoidance control, to effectively handle the safe interaction between vehicles, a nonlinear vector function $g(\cdot, \cdot)$ is introduced to describe the feature circle distance constraint between vehicles. This constraint function is closely related to the ego-vehicle state $x_{k/t}^i$ and the state vector of surrounding vehicles $x_{k/t}^j$ and serves as a key component in the optimization solution of Model Predictive Control (MPC).

Specifically, the reference trajectory of the ego-vehicle and the predicted trajectories of surrounding vehicles together constitute the basic parameters for solving the vehicle optimization problem at time t . Based on the optimal control sequence and the current system state, the controller infers the predicted state trajectory, and on this basis, establishes collision constraints for the vehicle over N prediction horizons. The nonlinear inequality constraint for multi-vehicle collision avoidance $g(x_{k/t}^i, x_{k/t}^j) \leq 0$ (where $j \in N_i(t)$) is designed to ensure the geometric separation between the ego-vehicle and surrounding vehicles within the time domain.

To accurately convert the rectangular geometric shape of the vehicle into computable mathematical constraints, the Feature Circle approximation method is adopted. The collision avoidance constraints between vehicles are defined as the following system of nonlinear inequalities, i.e.,:

$$g(x_k, x_k^{obs}) = \begin{bmatrix} D_g^2 - (x_{f,k} - x_{f,k}^{obs})^\top Q_c (x_{f,k} - x_{f,k}^{obs}) \\ D_g^2 - (x_{f,k} - x_{r,k}^{obs})^\top Q_c (x_{f,k} - x_{r,k}^{obs}) \\ D_g^2 - (x_{r,k} - x_{f,k}^{obs})^\top Q_c (x_{r,k} - x_{f,k}^{obs}) \\ D_g^2 - (x_{r,k} - x_{r,k}^{obs})^\top Q_c (x_{r,k} - x_{r,k}^{obs}) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

In the above equation, x_k and x_k^{obs} denote the states of the ego-vehicle and the obstacle vehicle (i.e., surrounding vehicle j), respectively, and Q_c is the weight matrix. This constraint essentially requires that the squared distance between the front/rear Feature Circles of the ego-vehicle and those of the surrounding vehicle must be greater than the square of the Feature Circle diameter D_g , thereby ensuring no physical contact.

The position coordinates of the centers of the Feature Circles— $x_{f,k}$ (front circle) and $x_{r,k}$ (rear circle)—are derived from the ego-vehicle's centroid state and heading angle, with their kinematic relationship defined as:

$$\begin{aligned} x_{f,k} &= x_k + d_f \cos(A_f x_k) + d_f \sin(A_f x_k), \\ x_{r,k} &= x_k + d_r \cos(A_r x_k) + d_r \sin(A_r x_k), \\ x_{f,k}^{obs} &= x_k^{obs} + d_f \cos(A_f x_k^{obs}) + d_f \sin(A_f x_k^{obs}), \\ x_{r,k}^{obs} &= x_k^{obs} + d_r \cos(A_r x_k^{obs}) + d_r \sin(A_r x_k^{obs}), \end{aligned} \quad (20)$$

Herein, d_f and d_r denote the offset distances of the centers of the vehicle's Feature Circles relative to the ego-vehicle's centroid, forward and backward along the longitudinal axis, respectively; D_g is the Feature Circle diameter. To extract heading angle information from the high-dimensional state vector for calculating geometric positions, sparse matrices A_f and A_r are introduced:

$$A_f = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & & \\ \vdots & \vdots & \ddots & \vdots & & \\ 0 & 0 & \cdots & 0 & & \end{bmatrix}_{6 \times 6}, \quad A_r = \begin{bmatrix} 0 & 0 & \cdots & 0 & & \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & & \\ 0 & 0 & \cdots & 0 & & \end{bmatrix}_{6 \times 6} \quad (21)$$

In the above definitions, all elements of A_f are zero except for the element at the first row and third column, which is set to 1; all elements of A_r are zero except for the element at the second row and third column.

The flexibility of this constraint model stems from its parameter configuration: when $d_f = d_r = 0$, the four quadratic inequalities in Equation (20) degenerate into the same form, i.e., the single-circle constraint model; when $d_f, d_r > 0$, the four distinct quadratic inequalities correspond to strict constraints that the front and rear double Feature Circles of the ego-vehicle do not come into contact with those of the surrounding vehicle pairwise, thereby achieving accurate coverage of the collision risk associated with the rectangular vehicle body. This nonlinear constraint is ultimately integrated into the optimization problem of Model Predictive Control (MPC) and acts synergistically with control saturation constraints, ensuring that the trajectory generated by the vehicle within the prediction horizon not only satisfies dynamic feasibility but also possesses rigorous collision avoidance capability.

For control input saturation constraints, the lower bound u_{min}^i and upper bound u_{max}^i of the control input for vehicle i are specified as:

$$u_{min}^i \leq u_{k|t}^i \leq u_{max}^i \quad (22)$$

To prevent optimization infeasibility in tight merging scenarios, the collision avoidance constraints (Equation (15)) are implemented as Soft Constraints using a heavy penalty function (Generalized Exterior Point). This ensures the solver always finds a solution, even if it requires temporarily penalizing a safety buffer violation. In extreme cases where the merging gap closes unexpectedly, the collision avoidance constraints (Equation (17))

will force the optimization to output a deceleration or braking command. If the trajectory becomes infeasible, a backup safety mechanism is triggered to bring the vehicle to a stop until a safe gap becomes available. Furthermore, to ensure the stability of the DMPC over an infinite horizon, a mathematical terminal constraint term J_f is incorporated into the cost function as a soft penalty:

$$J_f(x_{N_p|t}^i) = w_f \|x_{N_p|t}^i - x_{ref,N_p|t}^i\|^2 + \rho \cdot \max(0, g_{safe}(x_{N_p|t}^i)) \quad (23)$$

where ρ is a large penalty coefficient enforcing the terminal state to reside within the feasible safety region defined by Equation (19).

3.4.2. Neural Network-Based Policy Solving and Optimization

Although the nonlinear MPC model formulated in Section 3.4.1 can effectively handle vehicle dynamics and complex collision avoidance constraints, its online numerical solution often faces enormous computational burden in high-density traffic scenarios at ramps, making it difficult to meet the stringent (near-)real-time response requirements of autonomous driving systems. To address this issue, this section introduces how to use an offline-trained deep neural network (DNN) policy π_θ to approximate the optimization and solving process of online MPC, thereby resolving the real-time performance bottleneck. To ensure controller continuity, the input layer accepts a fixed number of nearest surrounding vehicles (e.g., $N_{neighbor} = 5$). A sorting mechanism based on Euclidean distance ensures that the most critical interaction targets are consistently retained in the input sequence across consecutive time steps. Prior to being fed into the network, all input state variables (e.g., velocity, gap distance) are normalized to the range $[-1, 1]$ using Min-Max Normalization. This standardization prevents gradient saturation and ensures balanced sensitivity across different state features. The detailed architecture of the policy network is presented in Table 4.

Table 4. Hyperparameters of the Neural Network Architecture.

Layer	Type	Neurons/Dimension	Activation Function
Input Layer	Fully Connected	State Dimension ($n_x \times N_{neighbor}$)	None
Hidden Layer 1	Fully Connected	256	ReLU
Hidden Layer 2	Fully Connected	128	ReLU
Output Layer	Fully Connected	Control Dimension (n_u)	Tanh (Scaled)

To construct a robust dataset for approximating the global optimal policy, we employed a ‘Teacher–Student’ learning paradigm. The high-precision nonlinear MPC (the ‘Teacher’) was deployed on offline servers to solve optimal control problems over 5000 episodes, generating a comprehensive set of state-action pairs.

The data generation protocol incorporated strict randomization to ensure the network’s generalization capability across diverse ramp merging scenarios:

1. **Scenario Initialization:** For each episode, the ego vehicle and surrounding vehicles were initialized with random states. The initial speeds were sampled from the operational range $[0, U_{limit}]$ (e.g., 0–30 m/s), and initial positions were randomized to cover various conflict levels.
2. **Traffic Density and Gaps:** To simulate realistic traffic variations, the arrival intervals of mainline vehicles were generated using a shifted negative exponential distribution, while ramp vehicle generation followed a Poisson distribution. This approach naturally created a wide range of inter-vehicle gaps and traffic density conditions, forcing the MPC to resolve complex merging conflicts.

3. Data Split: The collected dataset, consisting of the optimal trajectory sequences, was randomly partitioned into a training set (80%) and a validation set (20%). The validation set was strictly isolated from the training process to monitor the loss convergence and assess the generalization performance of the policy network π_θ .

Compared with real-time MPC, which has to shorten the prediction horizon or reduce the number of iterations due to constraints on on-board computing power, the trained neural network can learn the long-term temporal planning characteristics of the ideal MPC. More importantly, the neural network converts the complex optimization iteration process into matrix operation-based forward inference, increasing the decision frequency to the millisecond level. This effectively avoids insufficient vehicle response and abrupt changes in control actions caused by online computation delays, achieving real-time approximation of the globally optimal policy.

$$\min_{\theta} V(s) = \sum_{k=0}^{N_p-1} l(x_k, x_k^{ref}, \pi_\theta(s)[k]) \quad (24)$$

$$s.t. \quad x_{k+1} = f(x_k, \pi_\theta(s)[k]) \quad (25)$$

$$h_n(x_k, x_{n,k}^{obs}) \leq 0 \quad (26)$$

$$u_{min} \leq \pi_\theta(s)[k] \leq u_{max} \quad (27)$$

$$s_t = \left[x_t^T, \left(x_{t:t+N_p}^{ref} \right)^T, \left(x_{t:t+N_p}^{obs} \right)^T \right]^T \quad (28)$$

$$k=0,1,\dots,N_p-1; n=0,1,\dots,N_n-1 \quad (29)$$

At time t , given $s \in S$, the network outputs the action sequence $u_{0:N_p-1} \in U^{N_p}$ according to the policy $\pi: S \rightarrow U^{N_p}$. Herein, S denotes the vehicle observation state set, and U denotes the vehicle admissible control set. The output dimension of the network is determined by the product of the action dimension $|u|$ and the prediction horizon N_p , and is partitioned by the acting time step into $\pi_\theta(s)[0], \pi_\theta(s)[1], \dots, \pi_\theta(s)[N_p-1]$. In the single Multi-Layer Perceptron (MLP) network adopted in this study, the output layer is structured sequentially to match the total dimension defined by the product of action dimension $|u|$ and prediction horizon N_p . Specifically, the 1st to $|u|$ -th neurons correspond to the approximately optimal action executed at time step 0 within the virtual time horizon, while the $|u|+1$ -th to $2|u|$ -th neurons correspond to the action executed at time step 1, and this pattern continues for the entire sequence. The k -th step action $u_k \leftarrow \pi_\theta(s)[k]$ and the vehicle state x_k are input into the dynamical system $f: X \times U \rightarrow X$ (where X denotes the vehicle feasible state set), to recursively propagate to the vehicle state x_{k+1} at the next time step. d represents the distribution of observations. θ denotes the parameters of the to-be-optimized policy network π . The geometric layout of the road network and the initial configuration of the vehicles are illustrated in Figure 7.

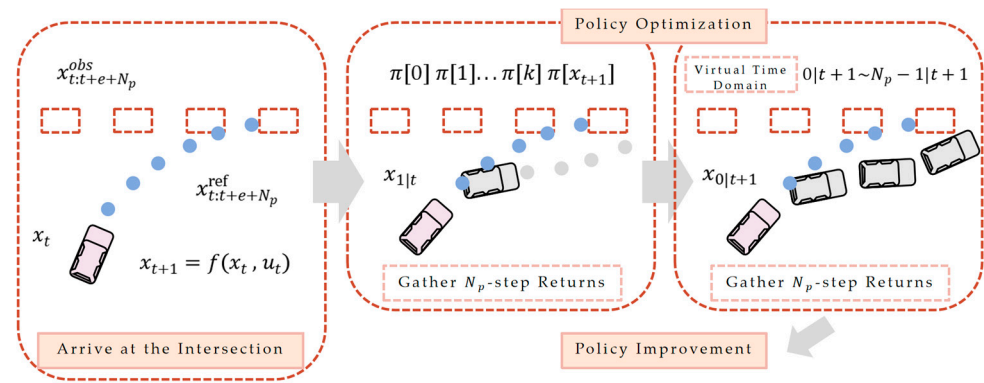


Figure 7. Flowchart of the Neural Network-Model Predictive Control (NN-MPC) Algorithm.

To provide a clear overview of the implementation, the step-by-step execution procedure of the proposed online distributed NN-MPC framework is summarized in Algorithm 2. This table details how the dynamic topology extraction, neural network acceleration, and optimization refinement are integrated within each control cycle.

Algorithm 2: Distributed NN-MPC with Dynamic Topology

Input: Ego state x_k , Neighbor states x_j , Reference x_{ref}

Output: Optimal control u_k

- 1: Initialize: Construct Dynamic Topology Graph $G(t)$
 - 2: for each time step $k = 0$ to $N - 1$ do
 - 3: **Step 1:** Neighbor Extraction
 - 4: Select $N_neighbor$ nearest vehicles based on Equation (8);
 - 5: **Step 2:** Neural Approximation (Hot-start)
 - 6: Predict initial guess $u_{init} = DNN(x_k, x_{ref}, x_{neighbors})$;
 - 7: **Step 3:** Optimization Refinement (if needed)
 - 8: Solve: $\min J(x_k, u_k)$ // Minimize Cost (Equation (10))
 - 9: Subject to:
 - 10: $x_{k+1} = f(x_k, u_k)$ // Vehicle Dynamics (Equation (11))
 - 11: $g_{safe}(x_k, x_{neighbor}) \leq 0$ // Feature Circle Collision Constraints (Equation (15))
 - 12: $u_{min} \leq u_k \leq u_{max}$ // Actuator Limits (Equation (19))
 - 13: Return u_k^*
 - 14: end for
-

This study approximates the collision avoidance constraints described in Equation (17) by adding a constraint violation penalty term to the objective function, using a differentiable activation function $\sigma(\cdot)$ for approximation:

$$\min_{\theta} V_p(s) = \sum_{k=0}^{N_p-1} l(x_k, x_k^{ref}, \pi_{\theta}(s)[k]) + \sum_{k=0}^{N_p-1} \sum_{n=0}^{N_p-1} \sigma(h_n(x_k, x_{n,k}^{obs})) \quad (30)$$

Notably, although the neural network policy π_{θ} can significantly improve computational speed, it is essentially a data-driven approximate solution. The output results theoretically cannot provide rigorous proof of hard constraint satisfaction like numerical optimizers. To ensure driving safety under extreme operating conditions, this study introduces a terminal Safety Check mechanism at the output of the policy network in engineering implementation. Specifically, when the control commands output by the network cause the predicted trajectory to violate the collision avoidance constraints defined in Equation (19), the system overrides the network output and strictly executes a rule-based Maximum Braking Strategy (AEB). This ensures fail-safe operation without incurring the

computational latency associated with numerical re-optimization. This hybrid architecture, utilizing the neural network for efficient planning and deterministic rules for safety boundaries, effectively balances the system's real-time performance and operational safety.

Although the specific loss curves are not plotted for brevity, the validation loss converged to the same order of magnitude as the training loss, indicating no significant overfitting. Furthermore, the high open-loop tracking accuracy (as shown in the simulation results) confirms that the DNN has learned the generalized control policy rather than memorizing specific trajectories.

3.4.3. Dynamic Interaction and Key Surrounding Vehicle Selection

The input dimension of neural networks is typically fixed. However, the traffic flow topology in ramp merging areas exhibits high time-variability, and the number of surrounding vehicles dynamically fluctuates over time. To address this dimension mismatch problem and enhance the system's adaptability and flexibility to complex scenarios, this study designs a dynamic surrounding vehicle selection Algorithm 3. By traversing the neighbor set $N_i(t)$ within the communication range, the algorithm calculates the potential conflicts between each surrounding vehicle and the ego-vehicle within the future prediction horizon.

Algorithm 3: Key Surrounding Vehicle Extraction Algorithm

Input: Ego-vehicle state x ; ego-vehicle reference trajectory x^{ref} ; prediction horizon N_p ; virtual surrounding vehicle states $x_{1:N_p}^{vir}|_t$; the set of all surrounding vehicles' predicted state machines $\{x_{1:N_p}^j|_t|a_{ij}(t)=1\}$; neighbor set $N_i(t)$

Output: Key surrounding vehicle's predicted states $x_{1:N_p}^{key}|_t$

```

1: Initialize the conflict set  $V_c \leftarrow \emptyset$  and the minimum index  $idx_{min} \leftarrow N_p$ 
2: for each neighbor  $j \in N_i(t)$  do
3:   for  $idx=0$  to  $N_p-1$  do
4:     if collision check function  $h(x_{idx}^{ref}, x_{idx}^j) > 0$  then
5:       Add tuple to set:  $V_c \leftarrow V_c \cup \{j, idx\}$ 
6:       break
7:     end if
8:   end for
9: end for
10: for  $(j, idx) \in V_c$  do
11:   if  $\cos \varphi_0(X_0^j - X_0) + \sin \varphi_0(Y_0^j - Y_0) \geq 0 \wedge idx < idx_{min}$  then
12:      $idx_{min} \leftarrow idx$ 
13:      $key \leftarrow j$ 
14:   end if
15: end for
16: if  $V_c = \emptyset$  then
17:    $x_{1:N_p}^{key}|_t \leftarrow x_{1:N_p}^{vir}|_t$ 
18: else
19:    $x_{1:N_p}^{key}|_t \leftarrow x_{1:N_p}^{key}|_t$ 
20: end if
21: return  $x_{1:N_p}^{key}|_t$ 

```

Algorithm 3 is mainly designed for ramp merging scenarios. It first undergoes two loops to identify all surrounding vehicles j that conflict with the ego-vehicle and their corresponding conflict time steps idx , which are then stored in the conflict set V_c . Second,

it extracts the vehicle with the earliest conflict time step idx . After training is completed, when deploying the network for multi-vehicle cooperative simulation, this algorithm is utilized to extract key surrounding vehicles or specify virtual surrounding vehicles. By selecting surrounding vehicles that may collide with the ego-vehicle and determining the key surrounding vehicle with the earliest potential conflict, the algorithm provides critical information for subsequent path planning and decision-making, thereby effectively addressing multi-vehicle interaction issues in complex traffic environments.

Under distribution shifts, rigorous formal verification of neural network-based controllers remains an open challenge. To mitigate risks, this study adopts a multi-layered safety strategy: (1) The training reward function incorporates heavy penalties for collision risks, enforcing the policy to learn conservative boundaries. (2) Strict kinematic constraints are imposed on the network's action space to ensure feasibility within physical limits. While a rigorous Lyapunov-based stability proof is beyond the scope of this applied study, extensive simulations cover diverse conflict scenarios to empirically validate closed-loop safety.

4. Experiments and Analysis

4.1. Experimental Setup, Parameters and Evaluation Metrics

To validate the effectiveness of the proposed distributed cooperative control framework, experiments were conducted under three typical bottleneck scenarios: ramp merging, ramp diverging, and intersection coordination. The simulation adopts a high-fidelity Automated Mechanical Transmission (AMT) vehicle dynamics model to reflect nonlinear characteristics and time delays. The experimental validation utilizes parameters of a standard C-Class passenger vehicle (see Table 5). While heavy-duty trucks exhibit different dynamics (e.g., larger inertia), the MPC formulation allows for parameter adaptation (e.g., updating mass m) in future extensions. This study focuses strictly on homogeneous passenger vehicle platoons.

The simulation environment assumes a fully connected V2X network, where vehicles share state information within the communication range (DSRC/4G). In the experimental setup, we assume an ideal communication environment to strictly isolate and evaluate the performance of the proposed cooperative mechanism. While we acknowledge that communication delays and packet losses are inevitable in real-world scenarios, the proposed NN-MPC framework is designed with inherent robustness against such disturbances. Specifically, the receding horizon mechanism (detailed in Section 3.4.1) updates trajectory planning at every time step. This allows the ego-vehicle to utilize the ‘prediction reuse’ capability: if a data packet is lost at time t , the system relies on the trajectory predicted at time $t-1$ (covering N_p future steps) to maintain safe operation. Future work will specifically quantify the boundaries of this robustness under varying packet loss rates.

The deep neural network was implemented using the PyTorch 1.2.0 framework. To verify the real-time application potential, the NN-MPC strategy was deployed on an on-board industrial computer. The control system takes the vehicle's state, static reference trajectory, and predicted trajectories of surrounding vehicles as inputs to solve for acceleration and steering angle commands in real-time. The industrial PC (Intel Core i7, 16 GB RAM) is equipped with an NVIDIA GPU to accelerate the process. The network parameters were updated using the Adam optimizer with an initial learning rate of 1×10^{-3} and a decay scheduler. The training process was conducted for a total of 5000 episodes, ensuring that the policy network converged to a stable solution with minimized loss.

Table 5. Heterogeneous Vehicle Parameters: Quantitative Analysis.

Vehicle Type	Mass m	Moment of Inertia I_z	Distance from Front Axle to Center of Mass l_f	Distance from Rear Axle to Center of Mass l_r	Equivalent Cornering Stiffness of Front Axle k_f
AMT	1412 kg	1537 kg/m ²	1.06 m	1.85 m	−128,916 N/rad

We established three distinct scenarios with varying traffic densities and conflict levels, including 2-vehicle and 4-vehicle interaction groups. To ensure a fair comparison, the benchmark MPC and the proposed NN-MPC were tested under identical traffic density and interference conditions (e.g., non-cooperative vehicle insertion). The performance is evaluated based on:

1. Computational Efficiency: Average inference time per step.
2. Safety and Conflict Resolution: Maintenance of minimum inter-vehicle distance and successful collision avoidance.
3. Ride Comfort: Magnitude of longitudinal deceleration and smoothness of steering angle adjustments.
4. Traffic Efficiency: Throughput continuity (avoiding forced stops).

4.2. Performance Analysis of Distributed Cooperative Trajectory Planning

This section evaluates the feasibility of the proposed Feature Point-based Path Generation (Section 3.3) and Dynamic Interaction Topology (Section 3.4.3) in complex merging and diverging scenarios.

4.2.1. Adaptive Merging at Ramp Intersections

As illustrated in Figure 8, four vehicles pass through the merging bottleneck cooperatively. At the initial phase ($t = 1.0$ s), the vehicles track the reference trajectories generated by Cubic Bézier Curves with high precision. As the black vehicle (ramp vehicle) approaches the conflict zone, the path feature point P_m^f is dynamically updated. Unlike fixed-path methods, the proposed algorithm generates a smooth merging trajectory (Figure 8e–h) that adapts to the real-time position of the mainline green vehicle.

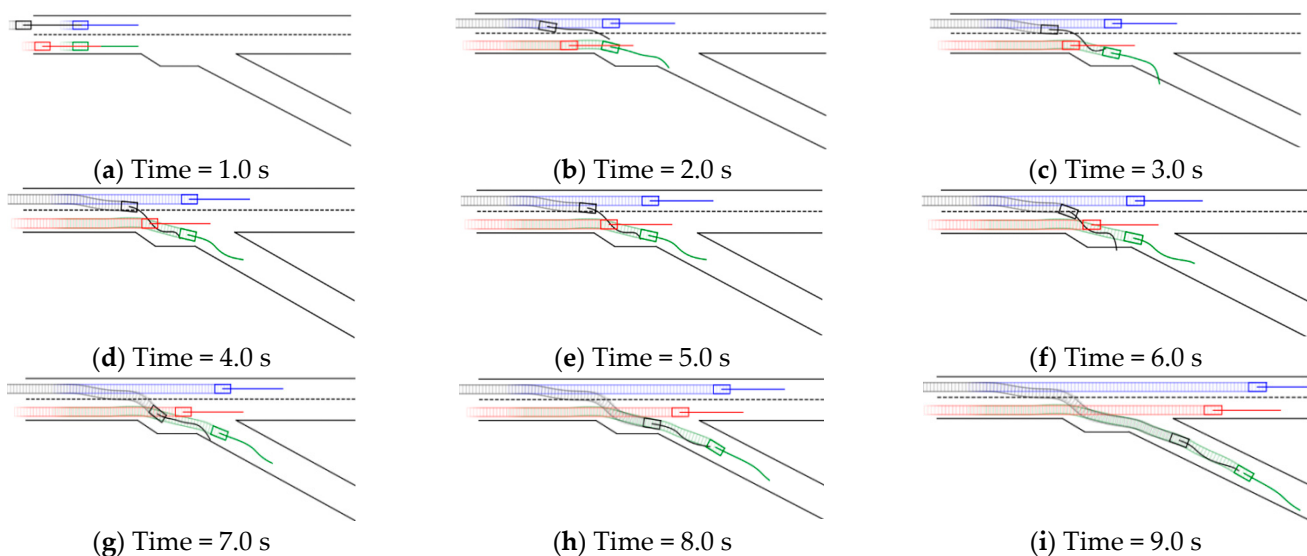


Figure 8. 4-Vehicle Cooperative Travel Trajectories at the Ramp Merging Intersection (Online Iterative Solution). (Red: Subject Vehicle; Green: Selected neighboring vehicles in the dynamic topology; Blue: Unselected surrounding vehicles).

Figures 9 and 10 illustrate the lateral and longitudinal control inputs for the merging scenario. The data reflects the specific advantages of the proposed distributed control: In the merging scenario, the baseline MPC often results in conflicts that affect the traffic efficiency of the ramp junction. In contrast, the proposed NN-MPC allows vehicles to adjust their merging angles in advance by utilizing the buffer lane. Quantitative analysis of velocity profiles shows that the longitudinal deceleration for merging vehicles exceeds 30%, yet the acceleration change is smaller and the steering angle is further reduced compared to the baseline MPC. This orderly merging flow significantly improves traffic efficiency while enhancing both safety and ride comfort.

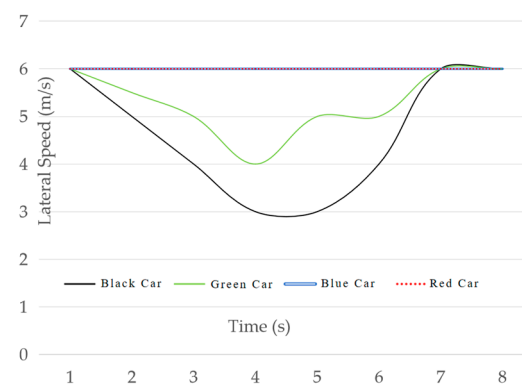


Figure 9. Lateral Speeds of Four Vehicles at the Ramp Merging Intersection.

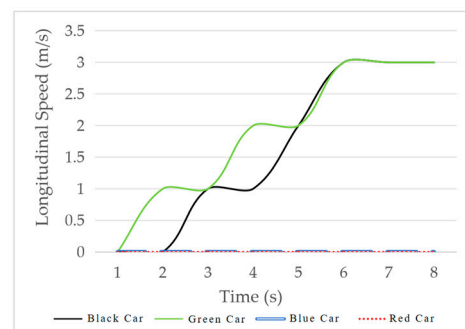


Figure 10. Longitudinal Speeds of Four Vehicles at the Ramp Merging Intersection.

4.2.2. Efficient Diverging at Ramp Intersections

In the diverging scenario (Figure 11), the red vehicle needs to exit the mainline. At $t = 2.0$ s, the red vehicle detects a potential collision with the straight-moving green vehicle, and the terminal Safety Check mechanism triggers a trajectory adjustment. Instead of abrupt braking, the red vehicle expands its turning radius and merges into the deceleration lane after the green vehicle passes ($t = 4.5$ s). The longitudinal speed curves (Figure 12a) exhibit synchronous fluctuations, indicating that the vehicles have successfully reached a Nash Equilibrium-like state through the cooperative cost function, balancing traffic efficiency and individual tracking performance.

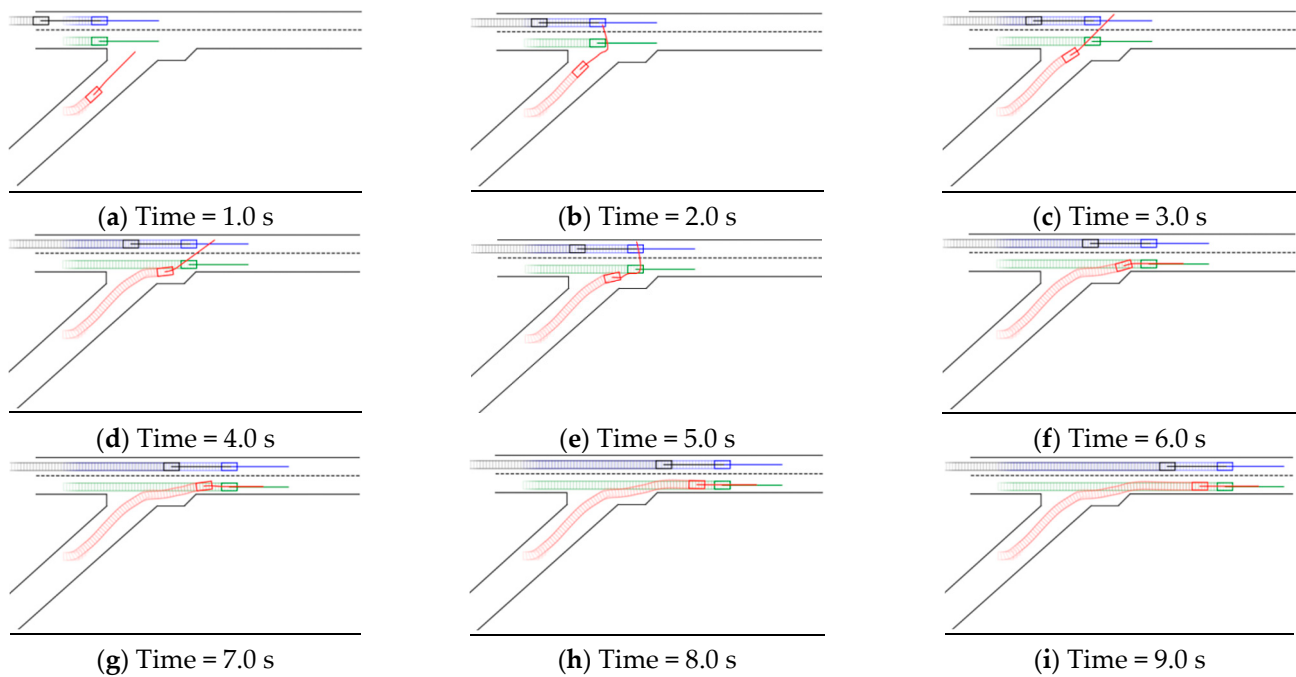


Figure 11. 4-Vehicle Cooperative Travel Trajectories at the Ramp Diverging Intersection (Online Iterative Solution).

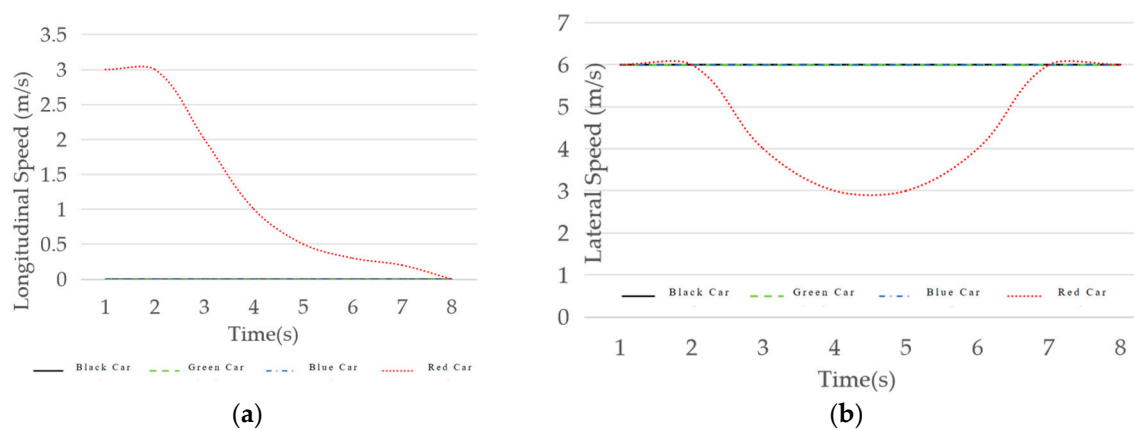


Figure 12. Longitudinal Speeds (a) and Lateral Speeds/Control Inputs (b) of Four Vehicles at the Ramp Diverging Intersection.

4.3. Comparative Experiments: NN-MPC vs. Real-Time Constrained MPC

To demonstrate the superiority of the proposed NN-accelerated strategy (Section 3.4.2), comparative experiments are conducted against a traditional nonlinear MPC as the benchmark. To simulate the resource-constrained characteristics of on-board computing environments, the benchmark MPC is restricted to a short prediction horizon and limited iteration steps, reflecting the “myopic” limitations of real-time solvers.

4.3.1. Overcoming Myopic Decision-Making (Two-Vehicle Conflict)

Baseline MPC Performance: In the ramp diverging conflict (Figure 13a), the traditional MPC forces a merge without yielding. The red vehicle attempts to cut in directly, completely disregarding the priority of the mainline vehicle. This forces the mainline vehicle to brake abruptly to avoid a collision. Such conflicts significantly degrade the traffic efficiency at the ramp intersection.

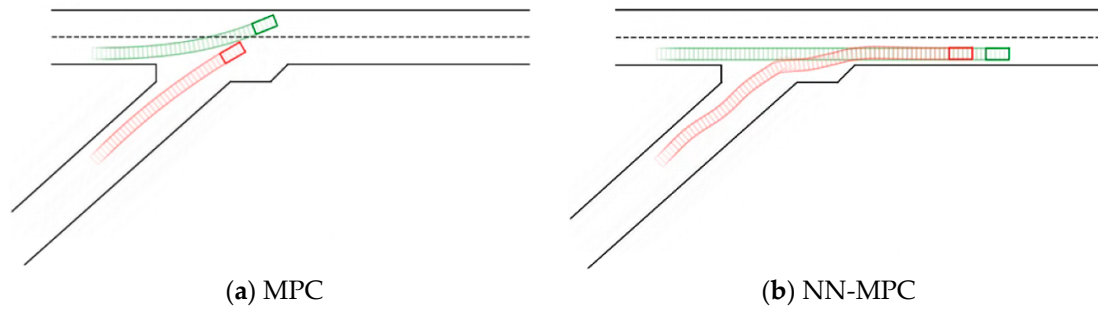


Figure 13. 2-Vehicle Cooperative Travel Trajectories at the Ramp Diverging Intersection.

NN-MPC Performance: In contrast, the NN-MPC (Figure 13b) demonstrates superior foresight. Upon detecting the conflict on the mainline, the red vehicle first adjusts its merging angle to yield to the green vehicle. It utilizes the buffer zone of the deceleration lane and merges only when the gap is sufficient. **Quantitative Comparison:**

1. **Ride Comfort:** The longitudinal speed change in NN-MPC is more stable, and the steering angle is smaller compared to the baseline.
2. **Safety:** The ordered merging prevents the dangerous cut-in observed in the MPC case, enhancing both passenger comfort and safety.

The difference in conflict resolution strategies between the two methods is clearly illustrated in Figure 14. In the merging scenario with a two-vehicle conflict, the vehicle controlled by the traditional MPC (red vehicle) attempts to merge directly without yielding to the main road vehicle. Consequently, the main road vehicle is forced to decelerate and yield early to satisfy safety constraints, which negatively impacts the traffic efficiency at the ramp interface.

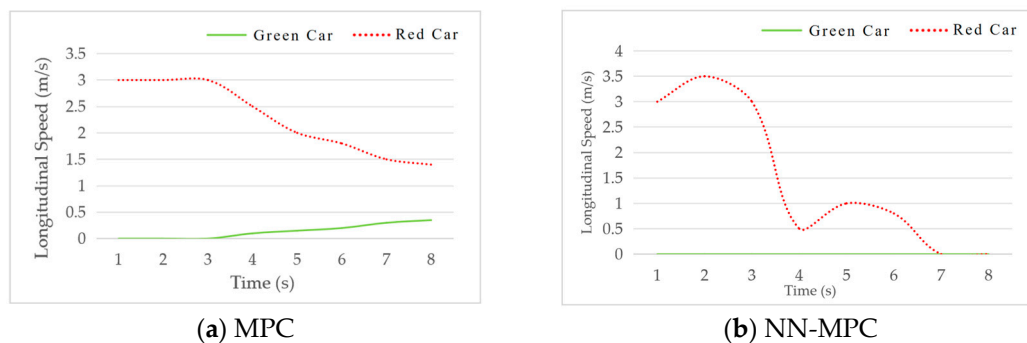


Figure 14. Longitudinal Speeds of 2-Vehicle Cooperative Travel at the Ramp Diverging Intersection.

In contrast, the proposed method demonstrates a superior cooperative behavior. As shown in Figure 14, when the ramp vehicle exits, it first adjusts its merging angle to proactively yield to the main road vehicle (green vehicle) upon detecting the conflict. It then utilizes the acceleration buffer lane to merge into the main road at an optimal timing. This strategy results in more stable longitudinal velocity and smaller steering angles, ensuring that the normal traffic flow on the main road is not disrupted. The resulting orderly merging process significantly enhances traffic efficiency, as well as ride comfort and safety.

4.3.2. Stability in High-Density Traffic (Four-Vehicle Conflict)

In the more complex four-vehicle scenario under high traffic density, the limitations of the baseline MPC become critical.

Baseline MPC Failure Analysis: As shown in Figure 15a, the red vehicle in the MPC group attempts to merge directly with an excessively large steering angle, failing to account for the continuous flow of mainline traffic.

1. **Impact on Throughput:** This behavior results in a forced stop-and-wait situation. The vehicle must wait for the mainline traffic to clear before merging, which interrupts the traffic flow. In scenarios with higher traffic volume, this behavior would directly lead to traffic congestion (traffic jams) and potential rear-end collisions.

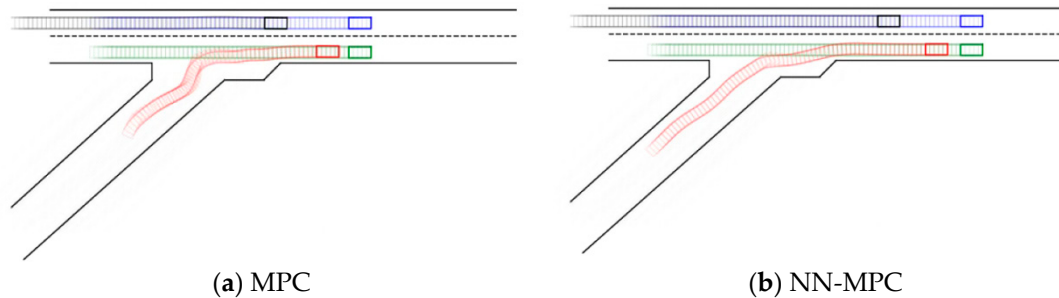


Figure 15. 4-Vehicle Cooperative Travel Trajectories at the Ramp Diverging Intersection.

NN-MPC Superiority: The NN-MPC method (Figure 15b) effectively utilizes the acceleration lane buffer.

1. **Trajectory Smoothness:** The vehicle performs an early lane change with a gradually adjusting steering angle.
2. **Efficiency Metrics:** The vehicle passes through the intersection without stopping (non-stop passing).
3. **Ride Comfort:** As seen in Figure 16b, the longitudinal acceleration is significantly smaller, and the reference trajectory switching is smooth.

This comparison confirms that while both methods can eventually complete the merge, the NN-MPC achieves it with better timing, smaller steering angles, and higher traffic throughput, avoiding the stop-and-go waves characteristic of the baseline approach.

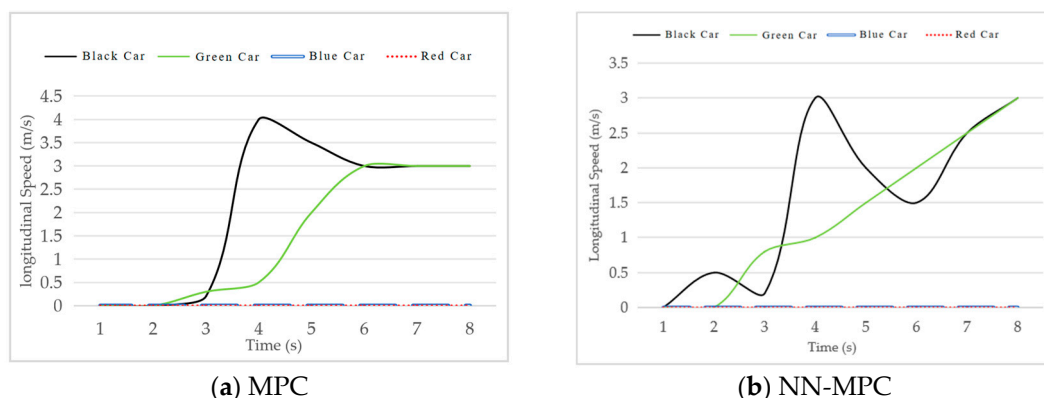


Figure 16. Longitudinal Speeds of 4-Vehicle Cooperative Travel at the Ramp Diverging Intersection.

Figure 17 illustrates the trajectories and velocity profiles of the four vehicles during the cooperative merging process. The difference in interaction logic is evident.

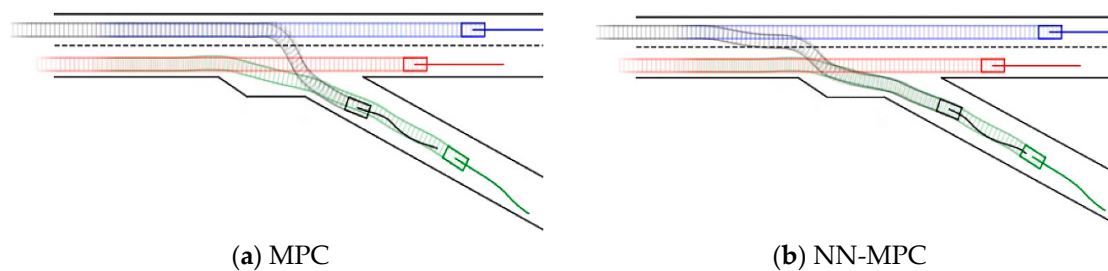


Figure 17. 4-Vehicle Cooperative Travel Trajectories at the Ramp Merging Intersection.

In the traditional MPC approach, the ego vehicle (red) attempts to merge aggressively with a large steering angle, disregarding the oncoming main road traffic. This forces the vehicles on the main road to decelerate or stop to avoid collision, significantly reducing the outflow efficiency and potentially causing traffic congestion in high-density scenarios.

In contrast, the proposed NN-MPC method demonstrates a highly cooperative strategy. As shown in Figure 17, upon detecting the conflict, the merging vehicle (red) proactively adjusts its steering angle and speed within the acceleration buffer lane to yield to the main road vehicle (green). It then merges into the main road smoothly at an optimal gap. This behavior ensures zero disruption to the main road traffic flow, resulting in smaller longitudinal acceleration and steering angle changes. The orderly merging process not only enhances traffic efficiency but also significantly improves safety and ride comfort.

4.3.3. Reliability and Safety Statistics

To assess the practical reliability of the proposed NN-accelerated strategy, we conducted a statistical analysis of the constraint violation rate over the entire set of 10,000 simulation steps across all test scenarios. The statistical results indicate that the raw control actions output by the policy network π_θ satisfied all safety and dynamic constraints in approximately 96.5% of the time steps. Consequently, the safety override mechanism (which triggers the fallback QP solver) was activated in only 3.5% of the cases. These overrides primarily occurred during critical conflict moments where the vehicle interactions were highly nonlinear and close to the safety boundaries. This low intervention rate confirms that the learned policy has effectively captured the safe driving patterns, while the terminal safety check serves as a necessary and efficient safeguard to ensure 100% collision-free operation.

4.3.4. Summary of Results

As summarized in Table 6, the NN-MPC demonstrates superior performance in computational efficiency by reaching millisecond-level solving speeds. Furthermore, the reduction in longitudinal acceleration and steering angles directly validates the improvement in ride comfort compared to the baseline MPC. This is quantitatively supported by the results in complex merging and diverging scenarios, where vehicles successfully executed significant speed adjustments (with longitudinal deceleration exceeding 30%) smoothly, thereby avoiding the abrupt braking or forced stops typically encountered in the baseline numerical MPC approach.

Table 6. Quantitative Performance Comparison.

Metric	Baseline MPC	Proposed NN-MPC
Solving Speed	Standard numerical solving	Millisecond level
Longitudinal Accel.	Higher fluctuations	Smaller
Steering Angle	Larger angle	Smaller
Traffic Efficiency	Impacted by conflicts	Greatly improved
Safety	Baseline safety	Orderly and no collisions

4.4. Experimental Conclusions

The comparative results confirm that the NN-MPC strategy overcomes the computational bottlenecks of traditional methods. Specifically, the comparison validates two critical hypotheses: (1) NN-MPC significantly outperforms the real-time constrained MPC (baseline) in terms of traffic efficiency and conflict resolution; and (2) Crucially, NN-MPC approximates the performance of the computationally expensive ‘Ideal MPC’ (Teacher) with negligible accuracy loss, effectively ‘distilling’ the global optimality into a real-time solver. This verifies the advantages of the proposed strategy in terms of computation speed and control quality. Experiments conducted on an industrial PC (Intel Core i7, 16 GB RAM, NVIDIA GPU) demonstrate the following key performance indicators:

1. **Computation Time:** The average inference time of the proposed NN-MPC is significantly lower than that required by the traditional optimization-based MPC solver. Specifically, the online inference phase takes less than 10 ms per control step on the tested hardware. This performance proves the system’s real-time feasibility, meeting the hard real-time constraints that are crucial for high-speed ramp merging scenarios.
2. **Control Performance:** The proposed hybrid architecture maintains the “far-sighted” planning capabilities of an ideal offline global planner while ensuring smooth network inference. Compared to the baseline MPC, the strategy effectively balances safety and riding comfort by reducing longitudinal deceleration during conflicts and eliminating forced stops, thereby significantly improving throughput.
3. **Safety:** By strictly enforcing kinematic constraints through the Feature Circle formulation and sharing predicted trajectories via the dynamic topology, the system ensures that minimum safe distances are never breached. This mechanism provides robust safety guarantees even in dynamic, multi-vehicle conflict environments.

5. Conclusions

This study proposes a distributed multi-vehicle cooperative trajectory planning and control framework for ramp merging and diverging scenarios, integrating Deep Neural Networks (DNNs) with Model Predictive Control (MPC). It aims to address the challenges of low traffic efficiency and insufficient safety in multi-vehicle cooperative travel under ramp intersection scenarios. Based on a hierarchical and progressive distributed control architecture, the method achieves flexible path planning and precise motion control of vehicles in complex traffic environments. By constructing a vehicle platoon interaction mechanism based on dynamic topology and a feature point-based Cubic Bézier Curve generation method, the system not only effectively reduces communication loads but also ensures the smoothness and spatiotemporal consistency of planned paths.

To solve the problem of excessive online computational burden of nonlinear MPC under high-density traffic flows, this study utilizes an offline-trained deep neural network to approximate the optimal control policy, converting the complex optimization iteration process into efficient forward inference. Notably, since this data-driven approximate solution theoretically cannot provide rigorous proof of hard constraint satisfaction, a terminal Safety Check mechanism and a dynamic surrounding vehicle selection algorithm are

introduced, realizing effective resolution of potential conflicts. Experimental validation was conducted in typical ramp merging and diverging scenarios, where the proposed NN-MPC strategy was compared with a real-time constrained MPC serving as the benchmark. The results demonstrate that the proposed strategy can successfully reproduce the planning capability of offline high-precision MPC while reducing the computation time to the millisecond level. It effectively overcomes the myopic decision-making problem of traditional real-time algorithms caused by computational delays or insufficient visual range. Compared with the benchmark method, NN-MPC exhibits superior smoothness in conflict resolution and traffic efficiency. Vehicles can smoothly complete merging and diverging without stopping by adjusting speed and steering angle in advance; quantitative validation confirms that this cooperative framework effectively smooths traffic flow, achieving an approximate 30% reduction in average travel delay compared to the non-cooperative baseline.

This confirms the engineering advantages of the hybrid architecture under dynamic high-density traffic flows—it not only guarantees the system’s real-time response capability to sudden incidents but also balances the safety and riding comfort of cooperative driving. However, the current modeling framework is limited by assumptions of ideal communication and nominal road friction conditions. The potential impact of packet loss, network delays, or adverse weather scenarios leading to reduced friction or sensor noise has not been explicitly tested. Future work will focus on these aspects by incorporating variable friction coefficients and sensor uncertainty models to validate the system’s robustness under extreme environmental conditions. Additionally, hybrid learning architectures, such as hybrid RL-PID [59], could be integrated to further compensate for model uncertainties in complex dynamic environments.

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

Funding: This work was supported by the Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport (ZHJTDSJ202402), Beijing Jiaotong University; Natural Science Foundation of Jiangsu Province (BK20231197, BK20220243); Science and Technology Program of Suzhou (SYG2024057, SYC2022078); Hubei Science and Technology Talent Service Enterprise Project (2023DJC084, 2023DJC195); Hubei Science and Technology Project (2024BAB087). Additionally, this work was sponsored by the Tsinghua-Toyota Joint Research Institute Interdisciplinary Program.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: Author Tingyang Zhang was employed by the company YTO Group Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

α	Learning rate of the neural network
ξ	Parameter of the Cubic Bézier Curve, $\xi \in [0,1]$
θ	Parameters (weights and biases) of the policy network
ρ	Penalty coefficient for safety constraint violation
φ	Heading angle (yaw) of the vehicle

ω	Yaw rate of the vehicle
π_θ	Policy network parameterized by θ
N_p	Prediction horizon steps
T_s	Sampling time step (s)
u_k	Control input vector at time step
x_k	State vector at time step
x_{ref}	Reference state vector
MPC	Model Predictive Control
NN-MPC	Neural Network–Model Predictive Control
DNN	Deep Neural Network

References

1. Carrillo-Zapata, D.; Milner, E.; Hird, J.; Tzoumas, G.; Vardanega, P.J.; Sooriyabandara, M.; Giuliani, M.; Winfield, A.F.T.; Hauert, S. Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection. *Front. Robot. AI* **2020**, *7*. <https://doi.org/10.3389/frobt.2020.00053>.
2. Farah, H.; van Beinum, A.; Daamen, W. Empirical Speed Behavior on Horizontal Ramp Curves in Interchanges in the Netherlands. *Transp. Res. Rec.* **2017**, *2618*, 38–47.
3. Papageorgiou, M.; Kotsialos, A. Freeway ramp metering: An overview. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 271–281.
4. Dong, J.; Chen, S.; Li, Y.; Du, R.; Steinfeld, A.; Labi, S. Space-weighted information fusion using deep reinforcement learning: The context of tactical control of lane-changing autonomous vehicles and connectivity range assessment. *Transp. Res. Part C* **2021**, *128*, 103192.
5. Chen, W.H.; Ren, G.; Cao, Q.; Du, Z.G.; Guo, W.W.; Huang, J.F. A game-theory-based approach to modeling lane-changing interactions on highway on-ramps: Considering the bounded rationality of drivers. *Mathematics* **2023**, *11*, 402.
6. Sun, S.C.; An, X.; Zhao, J.; Li, P.; Shao, H.P. Modeling and simulation of lane-changing management strategies at on-ramp and off-ramp pair areas based on cellular automaton. *IEEE Access* **2021**, *9*, 35034–35044.
7. Zhou, Y.; Ozbay, K.; Kachroo, P.; Chung, E. A supervised switching-mode observer of traffic state and parameters and application to adaptive ramp metering. *Transp. A Transp. Sci.* **2022**, *18*, 1178–1206.
8. Shi, Y.; Yu, H.; Guo, Y.; Yuan, Z. A collaborative merging strategy with lane changing in multilane freeway on-ramp area with V2X network. *Future Internet* **2021**, *13*, 123.
9. Daamen, W.; Loot, M.; Hoogendoorn, S.P. Empirical analysis of merging behavior at freeway on-ramp. *Transp. Res. Rec.* **2010**, *2188*, 108–118.
10. Calvi, A.; De Blasiis, M.R. Driver behavior on acceleration lanes: Driving simulator study. *Transp. Res. Rec.* **2011**, *2248*, 96–103.
11. Marczak, F.; Daamen, W.; Buisson, C. Empirical analysis of lane changing behavior at a free-way weaving section. *Traffic Manag.* **2016**, *3*, 139–151.
12. Kusuma, A.; Liu, R.; Choudhury, C.; Montgomery, F. Lane-changing characteristics at weaving section. In *Transportation Research Board 94th Annual Meeting*; Transportation Research Board of the National Academies: Washington, DC, USA, 2015; pp. 49–55.
13. Kondyli, A.; Elefteriadou, L. Driver behavior at freeway-ramp merging areas based on instrumented vehicle observations. *Transp. Lett.* **2012**, *4*, 129–142.
14. Wan, H.; Zhang, Y.; Guo, Z.; Li, Z.; Shu, X. Field Test Research on Vehicles Crossing the Lane Lines and Lane-Changing Locations Traffic Behavior in Multi-Lane Freeway Diverging and Merging Areas. *Dalian ICTE* **2015**, *2015*, 1434–1441.
15. Lyu, N.; Cao, Y.; Wu, C.; Xu, J.; Xie, L. The effect of gender, occupation and experience on behavior while driving on a freeway deceleration lane based on field operational test data. *Accid. Anal. Prev.* **2018**, *121*, 82–93.
16. Newell, G.F. A simplified car-following theory: A lower order model. *Transp. Res. Part B* **2001**, *3*, 195–205.
17. Dresner, K.; Stone, P. A multiagent approach to autonomous intersection management. *J. Artif. Intell. Res.* **2008**, *31*, 591–656.
18. Marinescu, D.; Čurn, J.; Bouroche, M.; Cahill, V. On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. In *Proceedings of the 2012 IEEE International Conference on Intelligent Transportation Systems (ITSC), Anchorage, AK, USA, 25 October 2012*; IEEE: New York, NY, USA, 2012; pp. 900–906.
19. Rios-Torres, J.; Malikopoulos, A.A. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1066–1077.

20. Vasirani, M.; Ossowski, S. Evaluating policies for reservation-based intersection control. In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence, Aveiro, Portugal, 24–26 September 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 39–50.
21. Lee, J.; Park, B.B.; Malakorn, K.; So, J.J. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. *Transp. Res. Part C Emerg. Technol.* **2013**, *32*, 193–206.
22. Kamal, M.A.S.; Imura, J.; Hayakawa, T.; Ohata, A.; Aihara, K. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1136–1147.
23. Dai, P.; Liu, K.; Zhuge, Q.; Sha, E.H.-M.; Lee, V.C.S.; Son, S.H. Quality-of-experience-oriented autonomous intersection control in vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1956–1967.
24. Min, H.; Fang, Y.; Wu, X.; Wu, G.; Zhao, X. On-ramp merging strategy for connected and automated vehicles based on complete information static game. *J. Traffic Transp. Eng. (Engl. Ed.)* **2021**, *8*, 582–595.
25. Wang, Z.; Zheng, Y.; Li, S.E.; You, K.; Li, K. Parallel optimal control for cooperative automation of large-scale connected vehicles via admm. In *Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 9 December 2018*; IEEE: New York, NY, USA, 2018; pp. 1633–1639.
26. Maestre, J.M.; Negenborn, R.R. *Distributed Model Predictive Control Made Easy*; Springer: Dordrecht, The Netherlands, 2014; Volume 69.
27. Jin, Q.; Wu, G.; Boriboonsomsin, K.; Boriboonsomsin, K.; Barth, M. Platoon-based multi-agent intersection management for connected vehicle. In *Proceedings of the 2013 IEEE International Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, 6–9 October 2013*; IEEE: New York, NY, USA, 2013; pp. 1462–1467.
28. Bian, Y.; Li, S.E.; Ren, W.; Wang, J.; Li, K.; Liu, H.X. Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10744–10754.
29. Uno, A.; Sakaguchi, T.; Tsugawa, S. A merging control algorithm based on inter-vehicle communication. In *Proceedings 1999 IEEE/IEEEJ/SAI International Conference on Intelligent Transportation Systems, Tokyo, Japan, 5–8 October 1999*; IEEE: New York, NY, USA, 1999; pp. 783–787.
30. Wang, Z.; Wu, G.; Boriboonsomsin, K.; Barth, M.J.; Han, K.; Kim, B.; Tiwari, P. Cooperative ramp merging system: Agent-based modeling and simulation using game engine. *SAE Int. J. Connect. Autom. Veh.* **2019**, *2*, 115–128.
31. Zhang, H.; Du, L.; Shen, J. Hybrid mpc system for platoon based cooperative lane change control using machine learning aided distributed optimization. *Transp. Res. Part B Methodol.* **2022**, *159*, 104–142.
32. Zhou, A.; Peeta, S.; Yang, M.; Wang, J. Cooperative signal-free intersection control using virtual platooning and traffic flow regulation. *Transp. Res. Part C Emerg. Technol.* **2022**, *138*, 103610.
33. Zheng, Y.; Li, S.E.; Li, K.; Borrelli, F.; Hedrick, J.K. Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Trans. Control. Syst. Technol.* **2016**, *25*, 899–910.
34. Katriniok, A.; Kleibbaum, P.; Joševski, M. Distributed model predictive control for intersection automation using a parallelized optimization approach. *IFAC-Pap.* **2017**, *50*, 5940–5946.
35. Alrifaa, B. *Networked Model Predictive Control for Vehicle Collision Avoidance*; RWTH Aachen University: Aachen, Germany, 2017.
36. GUO, J.H.; Luo, Y.G.; Li, K.Q. Adaptive fuzzy sliding mode control for coordinated longitudinal and lateral motions of multiple autonomous vehicles in a platoon. *Sci. China Technol. Sci.* **2017**, *60*, 576–587.
37. Xu, T.; Barman, S.; Michael, W. Levin. Smoothing-MP: A novel max-pressure signal control considering signal coordination to smooth traffic in urban networks. *Transp. Res. Part C* **2024**, *166*, 104760. <https://doi.org/10.1016/j.trc.2024.104760>.
38. Zhang, Y.; Cassandras, C.G. Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees. *Automatica* **2019**, *109*, 108563.
39. Qian, X.; Gregoire, J.; De La Fortelle, A.; Moutarde, F. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In *Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015*; IEEE: New York, NY, USA, 2015; pp. 3452–3458.
40. Zhao, X.; Wang, J.; Yin, G.; Zhang, K. Cooperative driving for connected and automated vehicles at non-signalized intersection based on model predictive control. In *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019*; IEEE: New York, NY, USA, 2019; pp. 2121–2126.
41. Mahbub, A.I.; Malikopoulos, A.A.; Zhao, L. Decentralized optimal coordination of connected and automated vehicles for multiple traffic scenarios. *Automatica* **2020**, *117*, 108958.
42. Luo, H.; Dridi, M.; Grunder, O. A branch-price-and-cut algorithm for a time-dependent green vehicle routing problem with the consideration of traffic congestion. *Comput. Ind. Eng.* **2023**, *177*, 109093.

43. Pei, H.; Feng, S.; Zhang, Y.; Yao, D. A Cooperative Driving Strategy for Merging at OnRamps Based on Dynamic Programming. *IEEE Trans. Veh. Technol.* **2019**, *99*, 1–11.
44. Yildirim, E.A.; Wright, S.J. Warm-start strategies in interior-point methods for linear programming. *SIAM J. Optim.* **2002**, *12*, 782–810.
45. Cagienard, R.; Grieder, P.; Kerrigan, E.C.; Cagienard, R.; Grieder, P.; Kerrigan, E.; Morari, M. Move blocking strategies in receding horizon control. *J. Process Control.* **2007**, *17*, 563–570.
46. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
47. Lin, Z.; Duan, J.; Li, S.E.; Ma, H.; Yin, Y.; Cheng, B. Continuous-time finite-horizon adp for automated vehicle controller design with high efficiency. In *Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 7 December 2020*, IEEE: New York, NY, USA, 2020; pp. 978–984.
48. Duan, J.; Liu, Z.; Li, S.E.; Sun, Q.; Jia, Z.; Cheng, B. Adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints. *Neurocomputing* **2022**, *484*, 128–141.
49. Lucia, S.; Karg, B. A deep learning-based approach to robust nonlinear model predictive control. *IFAC-Pap.* **2018**, *51*, 511–516.
50. Karg, B.; Lucia, S. Reinforced approximate robust nonlinear model predictive control. In *Proceedings of the 2021 23rd International Conference on Process Control (PC), Štrbské Pleso, Slovakia, 9 June 2021*; IEEE: New York, NY, USA, 2021; pp. 149–156.
51. Zhang, X.; Bujarbaruah, M.; Borrelli, F. Near-optimal rapid mpc using neural networks: A primaldual policy learning framework. *IEEE Trans. Control. Syst. Technol.* **2020**, *29*, 2102–2114.
52. Liu, Z.; Duan, J.; Wang, W.; Li, S.E.; Yin, Y.; Lin, Z.; Cheng, B. Recurrent model predictive control: Learning an explicit recurrent controller for nonlinear systems. *IEEE Trans. Ind. Electron.* **2022**, *69*, 10437–10446.
53. Guan, Y.; Ren, Y.; Sun, Q.; Li, S.E.; Ma, H.; Duan, J.; Dai, Y.; Cheng, B. Integrated decision and control: Toward interpretable and computationally efficient driving intelligence. *IEEE Trans. Cybern.* **2022**, *53*, 859–873.
54. Hou, Y.; Edara, P.; Sun, C. Modeling mandatory lane changing using Bayes classifier and decision trees. *IEEE Trans. Intell. Transp. Syst.* **2013**, *15*, 647–655.
55. Smirnov, N.; Liu, Y.; Validi, A.; Morales-Alvarez, W.; Olaverri-Monreal, C. A game theory-based approach for modeling autonomous vehicle behavior in congested, urban lane-changing scenarios. *Sensors* **2021**, *21*, 1523.
56. Kim, W.; Park, J.; Sung, Y. Communication in multi-agent reinforcement learning: Intention sharing. In *Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021*.
57. Zhang, J.; Zhang, Y.; Yao, E. A New Framework for Traffic Conflict Identification by Incorporating Predicted High-resolution Trajectory and Vehicle Profiles in a CAV Context. *Transp. Res. Rec.* **2025**, *2679*, 445–462.
58. Zhang, Y.; Liang, K.; Loo, B.P.Y. Measuring dynamic accessibility by metro system under travel time uncertainty based on smart card data. *J. Transp. Geogr.* **2025**, *127*, 104294.
59. Ben Hazem, Z.; Saidi, F.; Guler, N.; Altaif, A.H. A Hybrid Reinforcement Learning Framework Combining TD3 and PID Control for Robust Trajectory Tracking of a 5-DOF Robotic Arm. *Automation* **2025**, *6*, 56.

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