NEURAL MPC-BASED DECISION-MAKING FRAMEWORK FOR AUTONOMOUS DRIVING IN MULTI-LANE ROUNDABOUT

Yao Mu[1,*], **Zhiqian Lan**[2,*], **Chang Liu**[3], **Ping Luo**[1], **Shengbo Eben Li**[2,†]
[1]Department of Computer Science, The University of Hong Kong
[2]School of Vehicle and Mobility, Tsinghua University
[3]College of Engineering, Peking University
{ymu,pluo}@cs.hku.hk, {lanzq21@mails,lishbo@}.tsinghua.edu.cn
changliucoe@pku.edu.cn

ABSTRACT

Roundabouts are a popular alternative to traditional intersections due to their ability to reduce traffic collisions, but they pose challenges for autonomous vehicles due to their complex traffic situation. Previous attempts at decision-making in multi-lane roundabouts have been hindered due to the difficulty of accurately predicting the behavior of other traffic participants and the complexity of the overall traffic flow. To address the above challenges, this paper proposes a neural MPC-based decision-making framework for autonomous vehicles, in which multiple backup static paths are generate in real-time according to the road topology, and the decision-making problem is formulated as a series of parallel static path tracking problems with safety constraints, subject to dynamic surrounding vehicles. To overcome the uncertainty of traffic dynamics, we propose a neural model predictive control algorithm (NMPC), which learns a dynamics model with interaction data and solves the optimization problem with the gradient guidance via model predictive control. The path with the lowest cost is then chosen as the target path after solving all the constrained tracking problems and the corresponding action is chosen accordingly. To enhance computational efficiency, a critic network is used to approximate the constrained tracking cost and an actor network to approximate the control policy, reducing the burden of online solving. To evaluate the proposed framework, a multi-lane roundabout simulator is built to benchmark a real roundabout in Beijing and the proposed approach is tested with various densities of traffic flow. The results show that our method can successfully navigate the roundabout, perform lane change maneuvers safely and efficiently.

1 INTRODUCTION

Roundabouts have become a favored alternative to traditional intersections due to their ability to reduce severity of traffic collisions by eliminating left turns and allow vehicles to continuously move through the intersection, rather than having to wait for a stop sign or traffic light (Pérez et al., 2011). Generally, there are always multiple entrances and exits, and the driving process in roundabouts involves three stages: entering, circumnavigating, and exiting the roundabout. To enter the roundabout, drivers must yield to traffic already in the roundabout and then enter in a counterclockwise direction. Drivers then circumnavigate the roundabout, yielding to incoming traffic at each entry point, until they reach their desired exit. When exiting, drivers should signal their intention to leave the roundabout, and then wait for a gap in traffic before exiting. Although roundabouts provide convenience for human drivers, they pose significant challenges for decision-making and control of autonomous vehicles (Tian et al., 2018). In multi-lane roundabout the number of lane of different entrances and exits may be different and each entrance may have several vehicles entering and each exit may have vehicles exiting, thus the traffic flow in multi-lane roundabout has a high degree of

[*]Equal contribution.
[†]Correspondence to Shengbo Eben Li, email:lishbo@tsinghua.edu.cn

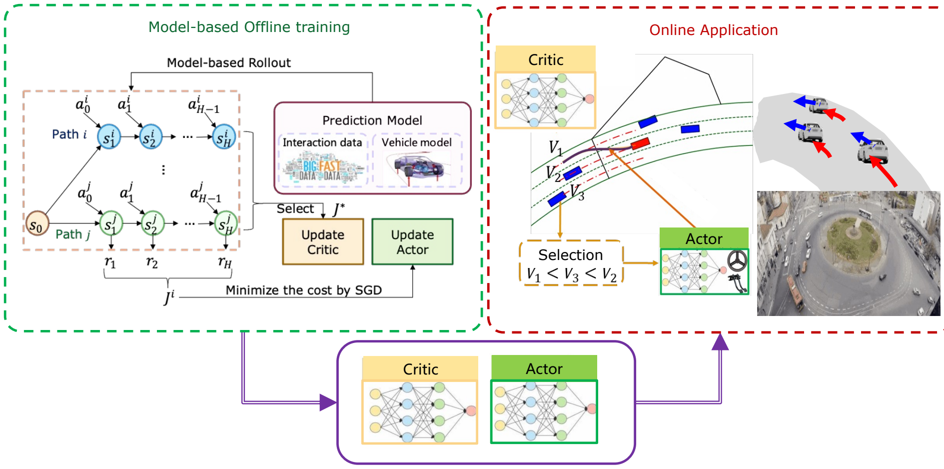

Figure 1: Overall framework of Neural MPC-based decision-making system for autonomous driving. The proposed framework is divided into two phases: model-based offline training and online application. In the offline training phase, the decision-making problem is formulated as a generalized constrained optimization problem that involves parallel static path tracking tasks under safety constraints in the presence of dynamic surrounding vehicles. A neural model predictive control (NMPC) solver is utilized to solve the optimization problem, which consists of a state trasition model, an actor network and a critic network. The state trasition model learns the state transition model and predicts future states according to the current actor's behavior. The actor is then optimized through stochastic gradient descent (SGD) by minimizing the cost of each constrained tracking problem. The critic learns the optimal cost of each constrained tracking task. In the online application phase, the critic is utilized to select the optimal target static path, while the actor outputs the corresponding actions. This seamless integration of the actor and critic allows for real-time decision-making while ensuring safety constraints are satisfied.

uncertainty (Abnili & Azad, 2021; Muhammad & Åstrand, 2019). Therefore, the intelligent vehicle requires quick and accurate reactions to drive the limited space and avoid collisions. Previous approaches require complex rules (Rastelli & Peñas, 2015) and modules to handle the diverse traffic situations, including lane selection and merging at intersections and unique inflow and outflow scenarios at each entrance and exit (Garcia Cuenca et al., 2019; García Cuenca et al., 2019). Despite these efforts, autonomous driving in multi-lane roundabouts still has a long way to go due to the complexity of traffic, road topology, and difficulty in predicting the behavior of other road participants.

The current decision-making framework in autonomous vehicles typically consists of prediction, planning, and control modules. The prediction module uses sensor data to anticipate the behavior of other traffic participants, such as pedestrians and vehicles. The planning module generates a safe and efficient target trajectory based on the static and dynamic traffic information, and the control module tracks the trajectory by sending commands to the vehicle's actuators. Early decision-making functions were largely based on expert rules (Urmson et al., 2009; Montemerlo et al., 2008). This framework designs behavior patterns for specific driving tasks such as lane maintenance, lane change, etc., using a finite state machine to switch and select driving behavior. With the rapid development of deep learning technology, end-to-end learning-based decision-making systems are also emerging. This architecture uses artificial neural networks as the carrier of control policy and directly learns a complex mapping from sensory input to control output. Bojarski et al. (2016) established a visual driving dataset, which maps image data to steering wheel angle, and realized lane maintenance based on convolutional neural networks. Lillicrap et al. (2015) used the DDPG algorithm to train a control policy in a simulation platform, achieving lane maintenance control. Duan et al. (2020) achieves decision and control in a two-lane highway scenario, using hierarchical RL method.

Model Predictive Control (MPC) as a typical receding horizon optimization method (Camacho & Alba, 2013; Rawlings, 2000) has gradually become a popular framework for autonomous control that can optimize the target trajectory and low level control together (Williams et al., 2018; Cesari et al., 2017; Liu et al., 2017). For example, Guan et al. (2022; 2021) proposed a integrated decision control framework for intersection scenario based on MPC framework. Conventional MPC used in previous works utilizes a mathematical model of the system to predict future behavior and determine control outputs that minimize a cost function that represents desired performance and constraints. However, there are several challenges in using MPC for multi-lane roundabouts. Firstly, the complex road topology and traffic environment make it difficult to formulate the whole driving process "entering-circling-exiting" to a unified optimal control problem. Secondly, the high uncertainty of vehicle behavior makes it challenging to accurately predict the future behavior of other road users and obstacles. Simplified mathematical model is hard to capture the behavior of surrounding traffic participators accurately. Finally, the process of solving large scale nonlinear optimization usually brings the high online computational burdens and leads to slow reaction issues.

To overcome the aforementioned challenges, we propose a neural MPC-based decision-making framework for the multi-lane roundabout scenario which casts decision-making problems under different situations into a generalized constrained optimization problem. The whole constrained optimization problem consists of a series of parallel static path tracking problems subject to the safety constraints with dynamic surrounding vehicles. The static paths are generated based on the road structure and typologies. To solve such constrained optimization problem with dynamic traffic flow in roundabout scenarios, we propose a Neural Model Predictive Control (NMPC) algorithm, which learns dynamics model with data-driven method to capture the characteristic of the current traffic flow, enabling accurate prediction of the behaviour of other traffic participants. Moreover, the gradients of state transitions given by the learned model are used as guidance to improve the efficiency of policy learning. We solve the constrained optimization problems for tracking each candidate static path with safety constraints separately, and choose the path with minimal optimal cost as the target path. The corresponding actions are then chosen. To further improve the computational efficiency, we train two networks to distill the knowledge obtained by solving the constrained optimization problem during training process. We learn a critic network to approximate the constrained tracking cost and an actor network to approximate the control policy. Thus, during testing process, the critic network can predict the constrained tracking cost for every backup static path, thus select the target static path with minimum cost, and the actor network can output the action directly without any online computing.

This framework has several advantages compared to conventional methods.

1). It has high online computing efficiency during the application. We can generate the static paths in real-time and two feed-forward networks are used to compute optimal target path and control command directly, which is also time-saving due to neural networks' extremely fast forward propagation.

2). It can be easily transferred to different traffic scenarios without a lot of human design. The static path planning module only uses static information of the road map. Thus, it's easy to develop static reference paths for different traffic scenarios (e.g., intersections, roundabouts, multiple lanes, interchange ramps, etc.). Besides, the constrained optimal control problem formulation is applicable for different kinds of dynamic traffic participants, including red lights, bicycles, and pedestrians.

2 NEURAL MPC-BASED DECISION-MAKING FRAMEWORK

We propose an neural MPC-based decision-making framework that consists of two parts, i.e., a static path planning module, a dynamic optimal tracking module and a neural model predictive control (NMPC) solver.

The static path planning module considers static traffic information like road structure and traffic signs to generate a set of candidate reference paths, along with their respective expected speeds. Unlike the planning modules in current autonomous driving systems, this module does not incorporate dynamic traffic participants in the scenario into its calculations. The generated static paths only indicate the permissible road area, determined by the shape and topology of the road. As a result, the computational cost of this approach is significantly lower compared to traditional trajectory planning methods.

The dynamic optimal tracking module is tasked with selecting and tracking the optimal path from the set of reference paths, taking into account dynamic traffic information in the scenario, such as the

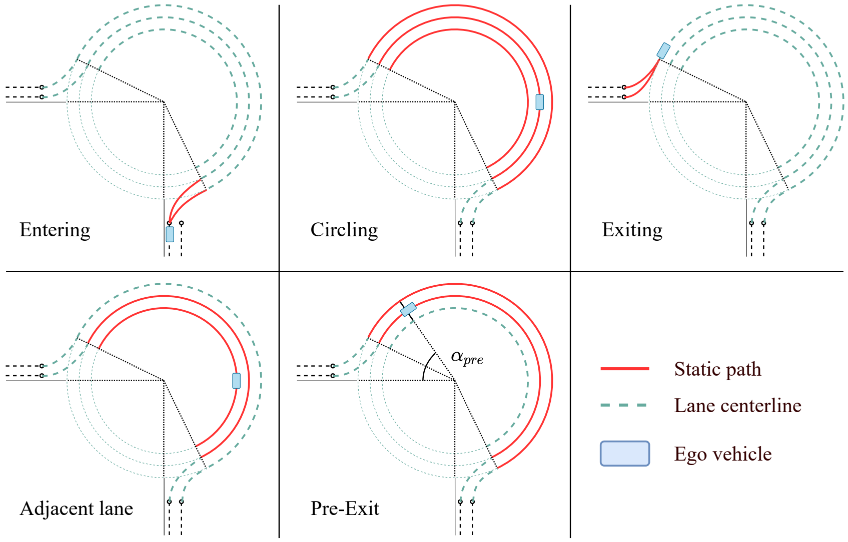

Figure 2: The generated static paths under different driving status.

presence of surrounding vehicles and pedestrians. For each candidate path, the module formulates a trajectory tracking problem with safety constraints of preventing collision with surrounding vehicles and road boundaries. Other driving objectives, such as avoiding harsh actions and maintaining relatively high speeds, are also reflected in the overall loss function. The module solves these optimal control problems in parallel and select the path with lowest optimal tracking cost as the target and then output the corresponding control commands.

The proposed Neural Model Predictive Control (NMPC) utilizes neural networks to enhance the precision of predicting the behavior of surrounding traffic participants. By integrating prior knowledge with interaction data, NMPC trains a prediction model that has superior capacity in anticipating traffic flow. The algorithm then uses the model to evaluate the total cumulative cost by rolling out H time steps from the current position. The model's state transitions are differentiable at every step, enabling the optimization of the policy through random gradient descent. This innovative combination of machine learning and control theory results in an efficient solution for learning the optimal policy in complex traffic flow.

2.1 Static Path Planning Module

The static path planning module generates multiple candidate paths based on static traffic information, such as road structure and topology. Each path is represented as a sequence of coordinate points, along with its expected velocity, serving as the primary driving guidance for the autonomous vehicle. The candidate paths are generated by equidistantly discretizing the centerline of lanes. In areas without defined lanes, such as intersections, virtual lanes are constructed to connect the allowable entrance and exit lanes, which are then discretized in the same manner. The expected velocity of each path is also assigned, taking into account the road region and traffic regulations. For example, the expected velocity of paths in intersections is set at a lower value, incentivizing the vehicle to slow down and avoid potential collisions.

This paper is focused on driving tasks in a 3-lane roundabout scenario, which can be divided into three stages: entering, circling, and exiting. The candidate static paths for each stage are illustrated in Figure 2. During the circling stage, the candidate static paths include the current lane of the ego vehicle and its adjacent lanes, allowing the vehicle to perform tactical lane changes to overtake other vehicles. In the entering and exiting stages, virtual lanes are constructed by linking the current lane of the ego vehicle to the exit lanes with Bezier curves, from which the static paths are generated.

To ensure driving safety and compliance with traffic rules, we incorporate two additional constraints in the generation of static paths, as illustrated in Figure 2. Firstly, the set of candidate paths is restricted to only include adjacent lanes, preventing continuous lane-changing behavior, which is prohibited by traffic rules. Thus, the outer lane is not included in the set of candidate paths for vehicles driving on the inner lane. Secondly, vehicles are not permitted to exit from the inner lane to prevent potential collisions and improve traffic efficiency. Therefore, vehicles that are within a predefined distance threshold from the exit are not provided with a static path of the inner lane.

Finally, a set of reference trajectory is generated based on the set of static path, serving as the input of the subsequent dynamic optimal tracking module. The reference trajectory, denoted as $\boldsymbol{\tau}(t), t \in \{0, 1, \dots, T-1\}$, is a temporal sequence of coordinates with T as length. It starts from the point on the static path nearest to the vehicle's current position and satisfies $\|\boldsymbol{\tau}(t) - \boldsymbol{\tau}(t-1)\|_2 = v_e \cdot \delta t$, where v_e is the expected speed of the static path and δt is the time step length.

It is worth noting that the static path planning module does not determine which path is optimal, neither does it consider the constraints of vehicle dynamics and the presence of other traffic participants. It only gives all the feasible driving area based on static traffic information, which is different from traditional trajectory planning methods. In addition, the static paths can be pre-processed and stored in an electronic map for faster retrieval, thereby improving real-time performance.

2.2 Dynamic Optimal Tracking Module

The dynamic optimal tracking module undertakes the task of selecting and tracking the optimal reference trajectory based on the reference trajectories and information of dynamic traffic participants. The module formulates an optimal control problem (OCP) for each candidate reference trajectory to track it under the constraints of road and surrounding traffic participants. After solving these OCPs, the trajectory with minimal tracking cost is selected as the tracking target and the corresponding solution as the control command.

2.2.1 Formulation of OCP

The mathematical form of the optimal control problem for tracking the ith trajectory $\boldsymbol{\tau}_i(t) = [p^{\text{ref}}_{x,t}, p^{\text{ref}}_{y,t}]^\top$ is as follows:

$$\begin{aligned} J_i &= \sum_{\tau=t}^{t+N} [J_x(\tau) + J_c(\tau)] + I_1 + I_2, \\ J_x(t) &= \left(x^{\text{ref}}(\tau) - x(\tau)\right)^\top \boldsymbol{Q} \left(x^{\text{ref}}(\tau) - x(\tau)\right), \\ J_c(t) &= u^\top(\tau) \boldsymbol{R} u(\tau), \end{aligned} \tag{1}$$

where $\boldsymbol{Q}, \boldsymbol{R}$ are positive-definite weighting matrices. The variables are defined as follows:

$$\begin{aligned} x(t) &= [p_{x,t}, p_{y,t}, v_t, w_t, \varphi_t, r_t]^\top, \\ x^{\text{ref}}(t) &= [p^{\text{ref}}_{x,t}, p^{\text{ref}}_{y,t}, v_e, 0, 0, 0]^\top, \\ u(t) &= [a_t, \delta_t]^\top, \end{aligned} \tag{2}$$

where $x(t)$ is the state of the vehicle, p_x, p_y are the position coordinates of the vehicle, v, w are longitudinal and lateral velocities, φ is the heading angle, r is the yaw rate. $u(t)$ is the control command as well as the optimization variable, where a is the acceleration and δ is the steering angle. The loss term J_x represents tracking cost to reference trajectory and J_c is added to prevent sharp turning and acceleration. In order to enable the autonomous vehicle to avoid the surrounding cars and prevent rushing out of the lane, another two collision constraints are incorporated into the overall loss function, namely I_1 and I_2. The form of vehicle-vehicle(V2V) collision constraint depends on the contour of vehicles, which are represented as rectangles.

Since it is hard to determine intersection of two rectangle based on solely center-to-center distance, we argue for the covering-circles method to represent the shape of vehicle and construct the V2V collision constraint. Specifically, as shown in Figure 3, three equal-sized circles are placed along the central axis of the vehicle with the second one located at the vehicle centroid. The centers of the circles are separated by a distance equal to their radius, which is set to 0.65 times the width of the vehicle.

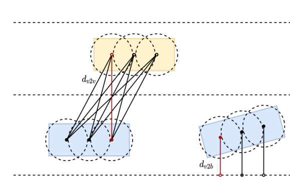

Figure 3: V2V and V2B collision constraints

To avoid collision with surrounding vehicles, the distance between each pair of covering circles of two cars should satisfy the following inequality:

$$\begin{aligned} & d_{j,k}^{\text{V2V}} \geq r_{\text{ego}} + r_{\text{surr}} + r_{\text{safe}}, \\ & j = \{1, 2, 3\}, \quad k = \{1, 2, 3\}, \end{aligned} \tag{3}$$

where d_{jk}^{V2V} is the distance between the jth circle of ego vehicle and the kth circle of the surrounding vehicle. r_{safe} is the safety margin of distance, set to be 0.1m in this paper. The vehicle-boundary(V2B) collision constraint is similar:

$$\begin{aligned} & d_{j,k}^{\text{V2B}} \geq r_{\text{ego}} + r_{\text{safe}}, \\ & j = \{1, 2, 3\}, \quad k = \{1, 2\}, \end{aligned} \tag{4}$$

where d_{jk}^{V2B} is the distance between the jth circle of ego vehicle and the $k-$th boundary.

After obtaining the above two constraints, we design the corresponding penalty function by applying a sigmoid function on the remaining distance:

$$I_1 = P_1 \sum_{i=1}^{N} \sum_{j=1}^{3} \sum_{k=1}^{3} \sigma \left[\rho \left(r_{ego} + r_{surr} + r_{safe} - d_{i,j,k}^{V2V}(\tau) \right) \right], \tag{5}$$

$$I_2 = P_2 \sum_{i=1}^{N} \sum_{j=1}^{3} \sum_{k=1}^{2} \sigma \left[\rho \left(r_{ego} + r_{safe} - d_{i,j,k}^{V2B}(\tau) \right) \right], \tag{6}$$

where P_1 and P_2 are the coefficients of each penalty term. The sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ can be seen as a differentiable version of the heaviside function, and ρ is a hyper-parameter which determines the intensity of safety-aware punishment. Given the candidate trajectory set $\{\tau_1, \tau_2, \dots\}$ and the corresponding optimal values $\{J_1^*, J_2^*, \dots\}$, the target trajectory τ_{i^*} is chosen to be:

$$i^* = \arg\min_k J_k^*. \tag{7}$$

The predictive tracking cost may vary based on the dynamic changes in the environment as it tracks a particular trajectory. Figure 4 demonstrates this phenomenon, where the predictive tracking cost rises when approaching the obstacle, suggesting the necessity and motivation of lane-changing. Figure 5 provides an exemplary illustration to clarify the functionality of the dynamic optimal tracking module, which tends to tracking the trajectory with lower predictive tracking cost.

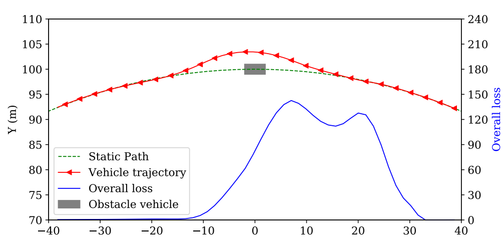

Figure 4: The curve of cost value when the vehicle approaches the obstacle while avoiding collision.

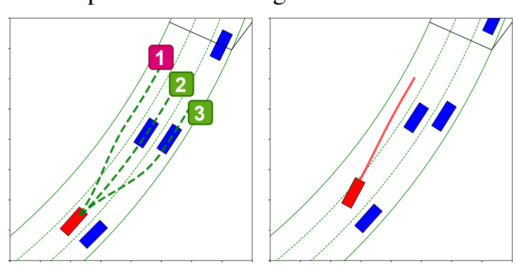

Figure 5: Example of Path selection. In this case, τ_1 has the lowest tracking cost, thus the ego vehicle (red) switches to the inner lane.

2.2.2 Local trajectory refinement

Although the reference trajectory starts from the closest point on the static path to the vehicle, there may still remain a notable gap between the vehicle and the reference trajectory, as shown in Figure 6.

$$\begin{aligned} & \boldsymbol{p}(t) = \boldsymbol{a}_0 + \boldsymbol{a}_1 t + \boldsymbol{a}_2 t^2 + \boldsymbol{a}_3 t^3, \\ & \boldsymbol{p}(0) = \boldsymbol{p}_0, \quad \dot{\boldsymbol{p}}(0) = \boldsymbol{v}_0, \\ & \boldsymbol{p}(L) = \boldsymbol{p}_l, \quad \dot{\boldsymbol{p}}(L) = \boldsymbol{v}_l, \end{aligned} \tag{8}$$

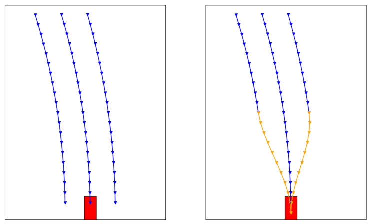

Figure 6: The static paths before and after local refinement

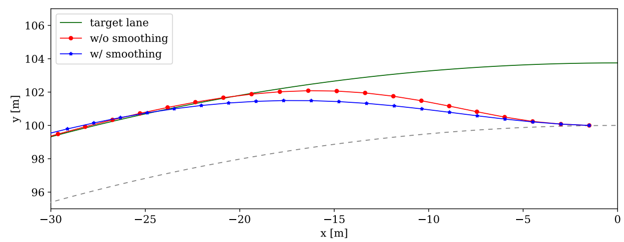

Figure 7: Vehicle performs lane-changing tasks under two kinds of reference trajectory.

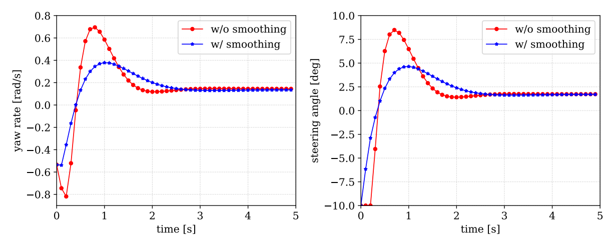

Figure 8: Curves of yaw rate and steering angle. The overshoot value of the steering control can be reduced from 389% to 179% by trajectory refine.

This may result in discontinuity in control output when the vehicle transits between different trajectories. To fix this issue, we introduce an additional module for local trajectory smoothing. For the trajectories with its projective distance exceeding a predefined threshold, their first L points are replaced by a locally generated point sequence starting from the current position of the vehicle. The local trajectory is represented by a third-order spline curve, where $t \in \{0, 1, \dots, L-1\}$. The coefficients of the cubic polynomial is determined to ensure the continuity of trajectory in both position and velocity. $\boldsymbol{p}_0, \boldsymbol{v}_0$ are current position and velocity vectors of the vehicle and $\boldsymbol{p}_l, \boldsymbol{v}_l$ are the position and velocity vectors of the Lth point in the origin trajectory which is to be refined. The coefficients are given by solving the above equations,

$$
\begin{aligned}
\boldsymbol{a}_0 &= \boldsymbol{p}_0, \quad \boldsymbol{a}_1 = \boldsymbol{v}_0, \\
\boldsymbol{a}_2 &= \frac{3(\boldsymbol{p}_l - \boldsymbol{p}_0) - (2\boldsymbol{v}_0 + \boldsymbol{v}_l)L \cdot \delta t}{(L \cdot \delta t)^2}, \\
\boldsymbol{a}_3 &= \frac{-2(\boldsymbol{p}_l - \boldsymbol{p}_0) + (\boldsymbol{v}_0 + \boldsymbol{v}_l)L \cdot \delta t}{(L \cdot \delta t)^3},
\end{aligned}
\tag{9}
$$

with $\delta t = 0.1$s as the time step length. We investigate the effects of local trajectory smoothing on vehicle control in a simulated lane-changing scenario. We utilized Model Predictive Control (MPC) to control the vehicle and track reference trajectories with and without the local smoothing operation. The results, as depicted in Figure 7, show that the implementation of local smoothing leads to a flatter vehicle trajectory. Furthermore, as illustrated in Figure 8, the curves of the yaw rate and steering angle exhibit a smoother profile, and the maximum yaw rate was reduced by 40%, indicative of enhanced driving comfort.

2.3 Neural MPC Solver

The Neural MPC solver consists of an actor network $\pi_\theta(\cdot)$, a critic network $v_\psi(\cdot)$, and a state prediction model $pre_\phi(\cdot)$. The actor network takes the current state and target path as inputs and outputs the action commands, while the critic network aims to fit the tracking cost for each path. The actor and critic networks need to be trained offline first. For each target path, we use the actor to sample H time steps of actions, and the state prediction model to predict the future state of the vehicle and other vehicles. The tracking cost and violations of constraints and their corresponding penalty costs are then calculated. The actor network can be updated using stochastic gradient descent (SGD). The critic network predicts the final cost of tracking each target path using the optimized actor, and is optimized using SGD. During online testing, the critic network is used to select the target static path with the minimum cost, and the actor network directly outputs the action. The state prediction model consists of a vehicle dynamics model of the ego vehicle and a state prediction model of surrounding vehicles. By combining these two models, a comprehensive state prediction model can be constructed to help predict the future behavior of all vehicles in a scene. To deal with the uncertainty of the environment, we use an RNN-based neural network (Medsker & Jain, 2001; 1999) to predict future states of the environment. As shown equation 10, an LSTM network (Hochreiter & Schmidhuber, 1997) is applied here to predict the next state of surrounding cars using historical information. In the prediction stage, information from the ego car-like steer angle, throttle, angular position, and information from surrounding cars like position and speed are stacked and sent into the LSTM encoder. A fully connected decoder is then used to predict the next state of surrounding cars from the hidden layer output of the LSTM network. Note that the residual connection is used here, and we predict the difference between the next state and the current state to reduce variance.

$$\begin{aligned} X_{\text{surr}}\,(t+1) &= X_{\text{surr}}\,(t) + pre_{\phi}(X(t), h_{t-1}; W, U, b) \\ X(t) &= \{X_{\text{surr}}\,(t), X_{\text{ego}}\,(t)\} \end{aligned} \tag{10}$$

3 Experiments

We construct the simulated scene to benchmark ROMA roundabout which is located in Changping District, Beijing based on SUMO (Lopez et al., 2018). Its main structure is a one-way circular road consisting of three lanes. External roadways converge to this roundabout from four orthogonal directions. The specifics and parameter values of the scenario are shown in Figure 11 and Table 3 in the Appendix. Besides, the values of parameters used in vehicle dynamics are depicted in Table 2. The whole driving task is divided into four sub-tasks: going straight, turning left or right, and making a U-turn. We conducted 100 independent experiments with random initialization for each of the four sub-tasks at three different levels of traffic density (40, 50, and 60 vehicles per thousand seconds). The reported results are the average performance across all experiments for each sub-task. The performance metrics include collision rate, driving-out-of-bound rate, comfort index, and average speed. In order to verify the effectiveness of our algorithm, we compared it against three different algorithms: a rule-based decision-making method provided by SUMO, MPC, and SAC, which is state-of-art model-free reinforcement learning algorithm. The results are reported in the Tabel 1. It turns out that our method outperforms or achieves parity with the best baseline approach across all performance criteria. We visualize the driving process of our method in a typical scenario with a dense traffic flow, as shown in Figure 9. It demonstrates that our algorithm can make lane-changing decisions reasonably under congested traffic conditions, while ensuring safety and improving traffic efficiency.

Algorithm	Collision	Out-of-Bound	Comfort index	Average Travel Speed
Neural MPC(Ours)	**0**	**0**	1.104	**14.74**
MPC (w/ data-driven)	3%	0	1.082	14.06
SAC	30%	24%	1.925	**15.37**
Rule Based	0	0	**0.846**	12.11

Table 1: Performance Comparison

4 Conclusion

This paper proposes a neural MPC-based decision-making (NMPC) framework for autonomous vehicles in multi-lane roundabouts. The proposed framework transforms the decision-making problem into a series of parallel static path tracking problems, while ensuring safety constraints are met. To address challenges arising from traffic uncertainty, the paper proposes a neural model predictive control algorithm that utilizes gradient guidance and an LSTM-based traffic participant behavior predictor to solve the constrained optimization problems. The results indicate that the proposed framework is successful in navigating the roundabout and performing lane change maneuvers safely and efficiently, outperforming the baseline methods.

5 Acknowledgement

Shengbo Eben Li and Zhiqian Lan are supported by Tsinghua University Initiative Scientific Research Program, and NSF China with U20A20334 and 52072213. Ping Luo is partially supported by the National Key R&D Program of China No.2022ZD0161000 and the General Research Fund of HK No.17200622.

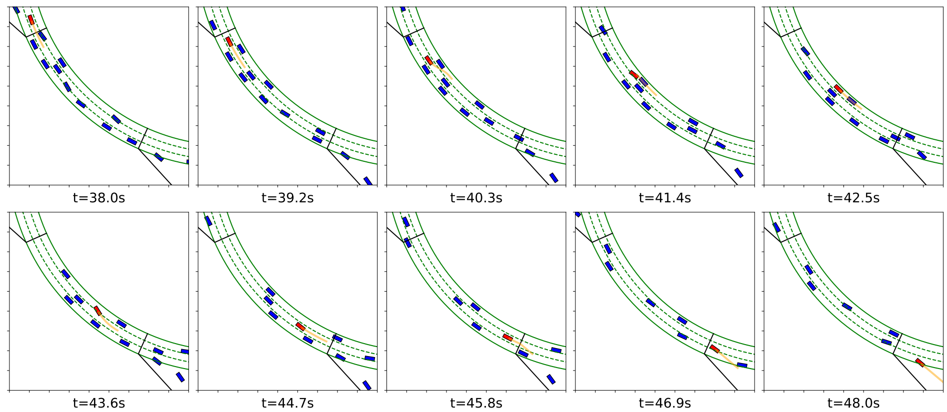

Figure 9: Visualization of a typical scenario in dense traffic, where the ego vehicle is represented by a red rectangle and the surrounding vehicles are represented by blue rectangles. The target reference trajectory for the ego vehicle is depicted as a yellow line in front of it. In order to pass a slow-moving vehicle in front, the ego vehicle first switched to the inner lane, completed the overtaking and then changed to the middle lane, and finally drove out of the roundabout.

References

Mehran Zamani Abnili and Nasser L Azad. On-line situational awareness for autonomous driving at roundabouts using artificial intelligence. *Journal of Machine Intelligence and Data Science (JMIDS)*, 2, 2021.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

Gianluca Cesari, Georg Schildbach, Ashwin Carvalho, and Francesco Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent transportation systems magazine*, 9(3):23–35, 2017.

Jingliang Duan, Shengbo Eben Li, Yang Guan, Qi Sun, and Bo Cheng. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data. *IET Intelligent Transport Systems*, 14(5):297–305, 2020.

Laura García Cuenca, Enrique Puertas, Javier Fernandez Andrés, and Nourdine Aliane. Autonomous driving in roundabout maneuvers using reinforcement learning with q-learning. *Electronics*, 8 (12):1536, 2019.

Laura Garcia Cuenca, Javier Sanchez-Soriano, Enrique Puertas, Javier Fernandez Andres, and Nourdine Aliane. Machine learning techniques for undertaking roundabouts in autonomous driving. *Sensors*, 19(10):2386, 2019.

Yang Guan, Yangang Ren, Haitong Ma, Shengbo Eben Li, Qi Sun, Yifan Dai, and Bo Cheng. Learn collision-free self-driving skills at urban intersections with model-based reinforcement learning. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3462–3469. IEEE, 2021.

Yang Guan, Yangang Ren, Qi Sun, Shengbo Eben Li, Haitong Ma, Jingliang Duan, Yifan Dai, and Bo Cheng. Integrated decision and control: toward interpretable and computationally efficient driving intelligence. *IEEE transactions on cybernetics*, 53(2):859–873, 2022.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Chang Liu, Seungho Lee, Scott Varnhagen, and H Eric Tseng. Path planning for autonomous vehicles using model predictive control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 174–179. IEEE, 2017.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pp. 2575–2582. IEEE, 2018.

Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

Naveed Muhammad and Björn Åstrand. Predicting agent behaviour and state for applications in a roundabout-scenario autonomous driving. *Sensors*, 19(19):4279, 2019.

Joshué Pérez, Vicente Milanés, Teresa De Pedro, and Ljubo Vlacic. Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts. *IFAC Proceedings Volumes*, 44 (1):13795–13800, 2011.

Joshué Pérez Rastelli and Matilde Santos Peñas. Fuzzy logic steering control of autonomous vehicles inside roundabouts. *Applied Soft Computing*, 35:662–669, 2015.

James B Rawlings. Tutorial overview of model predictive control. *IEEE control systems magazine*, 20(3):38–52, 2000.

Ran Tian, Sisi Li, Nan Li, Ilya Kolmanovsky, Anouck Girard, and Yildiray Yildiz. Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 321–326. IEEE, 2018.

Chris Urmson, Chris Baker, John Dolan, Paul Rybski, Bryan Salesky, William Whittaker, Dave Ferguson, and Michael Darms. Autonomous driving in traffic: Boss and the urban challenge. *AI magazine*, 30(2):17–17, 2009.

Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

A THE PSEUDO-CODE OF NEURAL MPC

Algorithm 1: Neural MPC Solver

Input: Current state s_t, backup static paths $[x^0_{ref}, x^1_{ref}, \ldots, x^N_{ref}]$
Output: selected Path x^*_{ref}action commands $a^{t+H}_{\tau=t}$
Initialize the actor network π_θ and critic network v_ϕ **for *each target path* x^i_{ref} do**
  for *each iteration* **do**
    Initialize cost $J = 0$.
    for *each time step* $\tau \in [t, t+1, \ldots, H]$ **do**
      Sample action with the actor network: $a_\tau \sim \pi_\theta(s_\tau)$.
      Predict future state of the vehicle s_{t+1} using the state prediction model $\phi(\cdot)$:
       $s_{t+1} = \phi(s_t, a_t)$.
      Calculate the current step cost c and update the overall cost: $J = J + c$.
    end
    Update the Actor network using SGD: $\theta = \theta - \alpha \frac{\partial J}{\partial \theta}$.
    Update the Critic network v_ψ using SGD: $\psi = \psi - \alpha \frac{\partial (v_\psi(s_t) - J)^2}{\partial \psi}$.
  end
  update the optimal cost: $J^*_i = J$
end
Select the path x^*_{ref} with the minimum cost from $x^0_{ref}, x^1_{ref}, \ldots, x^N_{ref}$

B IMPLEMENTATION DETAILS

This paper uses a 3-DoF vehicle model, with system dynamics governed by the following equation:

$$\dot{x} = f(x, u) = \begin{bmatrix} v \cos\varphi - w \sin\varphi \\ w \cos\varphi + v \sin\varphi \\ a + wr - \frac{k_f(w + l_f r - \delta v)\sin\delta}{mv} \\ \frac{k_f(w + l_f r - \delta v) + k_r(w - l_r r) - mvr}{mv} \\ r \\ \frac{l_f k_f(w + l_f r - \delta v) - l_r k_r (w - l_r r)}{I_{ZZ} v} \end{bmatrix} \tag{11}$$

The variables in 10 of vehicle dynamics are listed in Table 2.

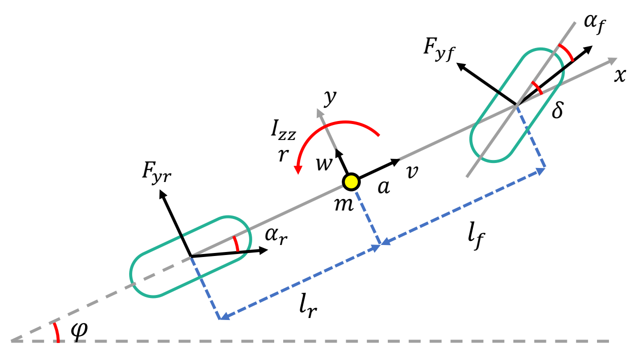

Figure 10: 3-DoF vehicle dynamical model

The detailed structure of ROMA roundabout scenario is shown in Figure 11. The structure parameters of ROMA roundabout which is used as the evaluation environment are listed in Table 3.

The LSTM network is applied in this paper to predict the next state of surrounding cars using historical information. In the prediction stage, information from the ego car-like steer angle, throttle,

Parameter	Description	Value
m	mass of vehicle	1412 kg
l_f	distance between C.G. and front axle	1.06 m
l_r	distance between C.G. and rear axle	1.85 m
k_f	front axle equivalent sideslip stiffness	-128915 N/rad
k_r	rear axle equivalent sideslip stiffness	-85943 N/rad
I_{ZZ}	yaw inertia of vehicle body	1536.7 kg $\cdot$ m^2

Table 2: Vehicle model parameters

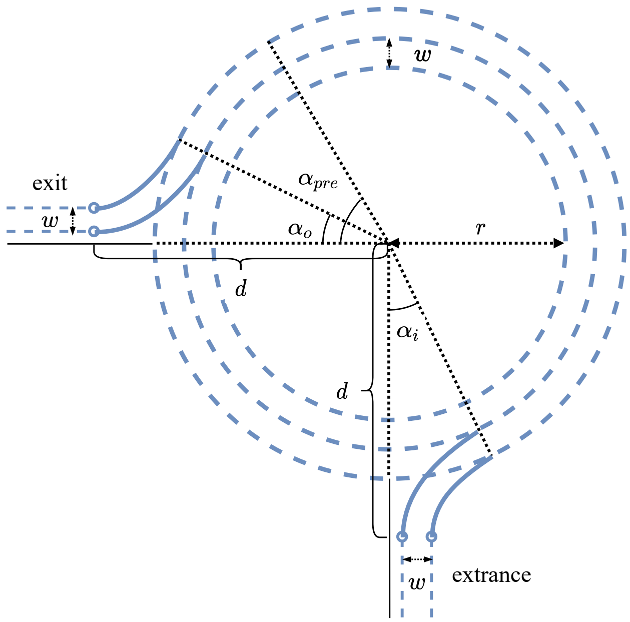

Figure 11: The structure and parameters of roundabout scene

angular position, and information from surrounding cars like position and speed are stacked and sent into the LSTM encoder. A fully connected decoder is then used to predict the next state of surrounding cars from the hidden layer output of the LSTM network. Note that the residual connection is used here, and we predict the difference between the next state and the current state to reduce variance.

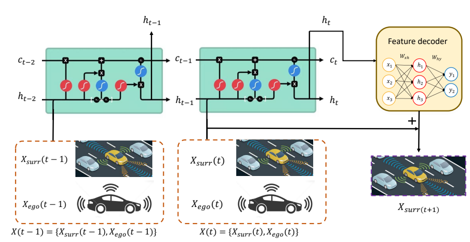

Figure 12: Surrounding traffic participants behavior prediction.

$$
\begin{aligned}
f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\
i_t &= \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \\
o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \\
h_t &= o_t \cdot \sigma_h\left(c_t\right)
\end{aligned} \tag{12}
$$

Parameter	Description	Value
w	lane width	3.75 m
r	radius of the inner lane	100 m
α_i	cut-in angle of the entering road	24 deg
α_o	cut-in angle of the exiting road	24 deg
α_{pre}	angle threshold of pre-exit area	60 deg
d	distance between origin and external road	135 m

Table 3: Roundabout parameters

where σ_g, σ_c, σ_h are all non-linear activation functions. Specifically, σ_g is the Sigmoid activate function, σ_c and σ_h are Hyperbolic Tangent function. h_t is the hidden state output by the recurrent neural network. The final state prediction $X_{\text{surr}}(t)$ is the output of feature transformation layer network $\psi(\cdot)$ which take the hidden state h_t as input.

C More Experimental Results

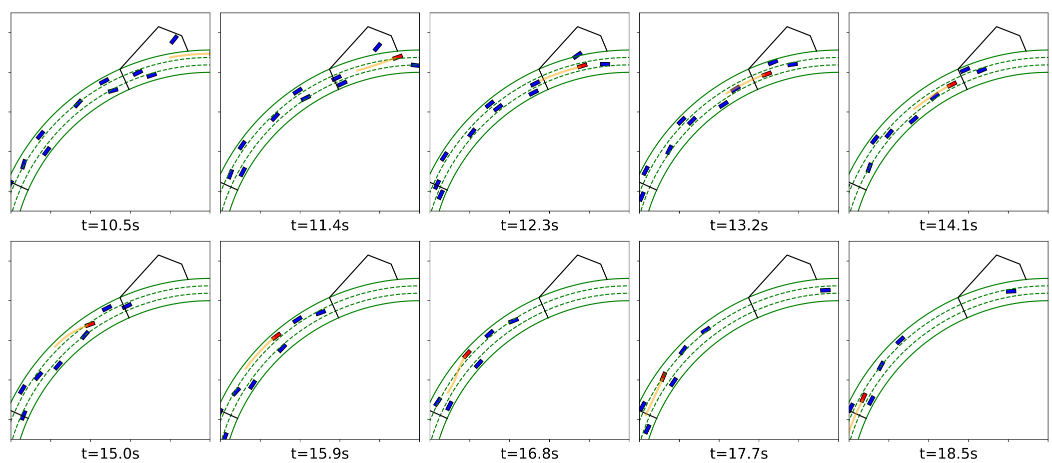

Figure 13: Visualization of lane change overtaking maneuver. The ego vehicle is represented by a red rectangle and the surrounding vehicles are represented by blue rectangles. The target reference trajectory for the ego vehicle is depicted as a yellow line in front of it. To avoid the merging traffic at an intersection, the ego vehicle made the decision to transit from the outer lane to the middle lane, and subsequently performed a lane change overtaking maneuver.