

A Conflicts-Free, Speed-Lossless KAN-Based Reinforcement Learning Decision System for Interactive Driving in Roundabouts

Zhihao Lin[✉], Zhen Tian[✉], Jianglin Lan[✉], Qi Zhang, Ziyang Ye[✉],
Hanyang Zhuang[✉], *Member, IEEE*, and Xianxian Zhao[✉]

Abstract—Safety and efficiency are crucial for autonomous driving in roundabouts, especially mixed traffic with both autonomous vehicles (AVs) and human-driven vehicles. This paper presents a learning-based algorithm that promotes safe and efficient driving across varying roundabout traffic conditions. A deep Q-learning network is used to learn optimal strategies in complex multi-vehicle roundabout scenarios, while a Kolmogorov-Arnold Network (KAN) improves the AVs' environmental understanding. To further enhance safety, an action inspector filters unsafe actions, and a route planner optimizes driving efficiency. Moreover, model predictive control ensures stability and precision in execution. Experimental results demonstrate that the proposed system consistently outperforms state-of-the-art methods, achieving fewer collisions, reduced travel time, and stable training with smooth reward convergence.

Index Terms—Roundabout, interactive driving, reinforcement learning, autonomous vehicle, Kolmogorov-Arnold network.

I. INTRODUCTION

ROUNDABOUT designs vary by city scale [1], but typically feature a central island that vehicles must circulate around—clockwise or counterclockwise—facilitating smoother traffic flow and reducing interaction complexity [2]. As urban roadways evolve, roundabouts have improved traffic distribution and increased road capacity [3]. While they generally present fewer conflicts than other intersections [4], safety concerns intensify in high-traffic conditions due to a greater crash risk [5], especially in mixed traffic

with both human-driven vehicles (HDVs) and autonomous vehicles (AVs).

Understanding human driving behavior in roundabouts, especially during entering, circulating, and exiting, is a key research focus. Merging and exiting require AVs to interact with surrounding HDVs, interpreting their intentions to make optimized decisions. Lane changes demand careful monitoring of HDVs and precise control to follow planned trajectories. Therefore, AVs navigating roundabouts must select lanes appropriately, monitor their environment, avoid collisions, and maintain precise control.

Research on autonomous driving in roundabouts has advanced from basic navigation to complex mixed-traffic scenarios. AVs reduce safety incidents caused by human errors like fatigue and distraction [6] and can make faster, optimal decisions [7]. They also enhance roundabout capacity [8]. With AVs expected to exceed 50 million on the road by 2030 [9], modern roundabouts are increasingly designed to support safe AV-HDV interactions [10], [11]. Control strategies for connected autonomous vehicles prioritize safety and efficiency [12], while current roundabout designs improve traffic flow and safety [13].

Control methods for autonomous driving in roundabouts have gained significant attention, with Model Predictive Control (MPC) and game theory being prominent model-based approaches. Game theory models decision-making by balancing safety, efficiency, and comfort [14], but often relies on simplified environments and struggles with real-world complexity [15]. Other model-based frameworks are limited by simplistic roundabout designs, few vehicles, and focus on abnormal cases [16]. While MPC effectively handles vehicle dynamics and safety constraints, current approaches still face challenges in complex real-world roundabouts [17], [18].

Learning-based methods, including machine learning and deep reinforcement learning (DRL), show strong potential for complex roundabout driving. Machine learning has been applied to AV-HDV interactions [19], but often requires extensive labeled data and struggles with generalization. DRL enables exploration of strategies in complex environments [20] and balances safety and efficiency in dense traffic [21]. Popular DRL algorithms include Deep Deterministic Policy Gradient (DDPG) [22], Proximal Policy Optimization (PPO) [23], and deep Q-learning (DQN) [24]. DDPG suits continuous actions but is less effective for discrete decisions like

Received 15 August 2024; revised 15 December 2024 and 25 May 2025; accepted 6 June 2025. Date of publication 25 June 2025; date of current version 21 October 2025. This work was supported in part by China Scholarship Council Ph.D. Scholarship for 2023–2027 under Grant 202206170011, in part by the Leverhulme Trust Early Career Fellowship under Grant ECF-2021-517, in part by U.K. Royal Society International Exchanges Cost Share Program under Grant IEC\NSFC\223228, and in part by the Sustainable Energy Authority of Ireland (SEAI) under RD&D Award 22/RDD/776. The Associate Editor for this article was H. H. Song. (Zhihao Lin and Zhen Tian contributed equally to this work.) (Corresponding author: Xianxian Zhao.)

Zhihao Lin, Zhen Tian, and Jianglin Lan are with the James Watt School of Engineering, University of Glasgow, G12 8QQ Glasgow, U.K.

Qi Zhang is with the Faculty of Science, University of Amsterdam, 1098 XH Amsterdam, The Netherlands.

Ziyang Ye is with the School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, SA 5005, Australia.

Hanyang Zhuang is with the University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China.

Xianxian Zhao is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8 Ireland (e-mail: xianxian.zhao@ucd.ie).

Digital Object Identifier 10.1109/TITS.2025.3578279

roundabout driving [25]. PPO achieves safe, efficient strategies in dense traffic [13] but its on-policy learning limits use of historical data, which is vital in complex environments like roundabouts [26]. DQN has been effectively applied to traffic simulations, excelling in tasks like intersection management relevant to roundabout navigation [27]. Its discrete action framework suits lane selection without action discretization, and experience replay enhances learning efficiency using past data. Additionally, DQN is computationally efficient [28]. The recently proposed Kolmogorov-Arnold Network (KAN) outperforms traditional multi-layer perceptrons by replacing linear layers with adaptive B-spline functions, enabling flexible feature extraction and improved generalization across diverse environments [29], [30].

To solve the complex driving in roundabouts, this paper proposes to integrate KAN with DQN (K-DQN) to enhance the decision-making and learning capabilities of AVs in complex roundabout scenarios [31]. The K-DQN leverages the advantages of both DQN and KAN, enabling AVs to learn robust and efficient driving strategies through interaction with the environment. For conflict-free driving, we introduce an action inspector applied to time to collision (TTC) [32] to assess the relative collision risks between the AV and other HDVs. By replacing dangerous actions that may cause collisions with safe actions, our proposed method can decrease the ego vehicle collision rates with neighboring vehicles (NVs) during training. For proper lane selection, we introduce a route planner that considers the number of HDVs and the available free-driving space in each lane. For precise control of planned trajectories, we implement MPC to allow the AV [33] to navigate with precision and robustness.

The main contributions of this paper are as follows:

- We propose a novel K-DQN to enhance AV decision-making in complex roundabouts. Compared to the traditional neural networks, the unique spline-based activation functions of KAN enable more precise environmental learning and decision-making, resulting in better training convergence, lower collision rates, and higher average speeds.
- Unlike prior methods that treat safety and efficiency separately, we introduce an integrated approach combining an action inspector and route planner. By merging TTC-based safety checks with density-aware lane selection, our method significantly reduces collisions while improving driving efficiency across diverse traffic conditions, compared to benchmarks.
- We enhance traditional DRL by integrating MPC with K-DQN, translating planned actions into safe, smooth controls. Our integrated solution adeptly manages diverse roundabout traffic flows, showing improved speed stability and efficiency over current benchmarks.
- We present mathematical analysis and experimental demonstrations to substantiate the superior performance of our K-DQN over traditional DQN methods. Extensive simulations confirm robustness and efficiency of our approach and its advantages over benchmarks.

The rest of this paper is organized as follows: Section II presents the problem statement and system structure;

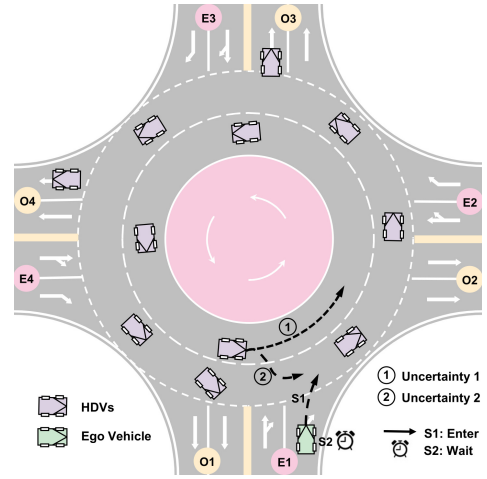


Fig. 1. A four-entrance, four-outlet, two-lane roundabout with the first collision scenario involving AV uncertainty.

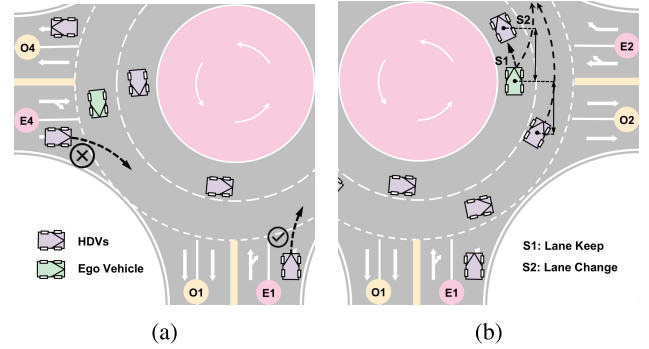


Fig. 2. Roundabout potential collision scenarios (a) and (b).

Section III describes the enhanced K-DQN; Section IV introduces the action inspector and route planner; Section V presents the MPC design; Section VI provides the simulation results with analysis; Section VII draws the conclusion.

II. PROBLEM STATEMENT AND SYSTEM STRUCTURE

The previous section highlighted the focus on AV-HDV interaction in roundabouts. However, integrating decision-making, path planning, and control remains challenging due to the complexity of roundabout scenarios. Unlike straight or other curvy roads, roundabouts present unique challenges in making safe and efficient decisions due to their complex network of entrances and outlets. HDVs can be randomly and densely distributed along both the inner and outer boundaries of roundabouts, frequently resulting in unexpected outcomes such as conflicts and inefficient driving. As shown in Fig. 1, the roundabout has four ports, each split into an entrance (right) and an outlet (left). HDVs' unpredictable maneuvers and unknown destinations pose challenges for AVs to ensure both safety and efficiency, defined here as minimizing travel time to the outlet. This study considers a signal-free, double-lane roundabout with both AVs and two types of HDVs: those already inside and those merging in.

Three potential collision situations have been identified in Fig. 1, Fig. 2(a), and Fig. 2(b), respectively. In Fig. 1, an AV

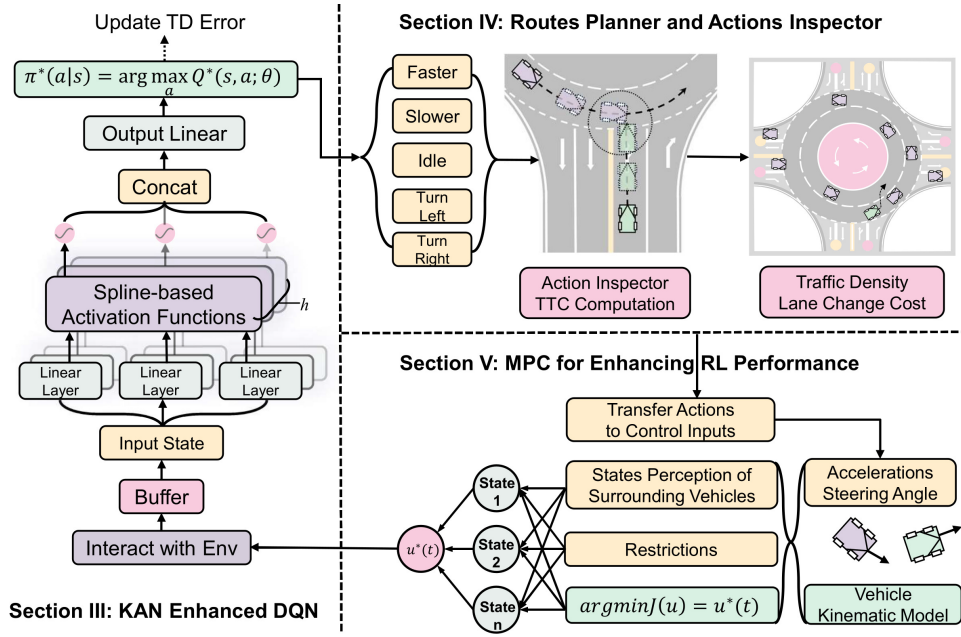


Fig. 3. The KAN-based, conflict-avoidance, and proper-lane-detection DRL system.

at an entrance must decide whether to wait for an approaching HDV in the outer lane—potentially increasing delay—or merge immediately, risking a collision due to limited distance and uncertain HDV speed. Fig. 2(a) presents the reverse case: the AV is approaching an entrance while an HDV attempts to merge, creating a conflict if both proceed simultaneously. In Fig. 2(b), the AV intends to exit from the inner lane but encounters an HDV in the adjacent outer lane. It must choose between overtaking with risk, following to reduce conflict, or delaying the lane change. These scenarios highlight the complexity and uncertainty AVs face in interactive decision-making with HDVs in roundabouts.

This paper focuses on ensuring the safety and efficiency of AVs navigating roundabouts under varying HDV densities and traffic flows. The proposed system, illustrated in Fig. 3, addresses this by integrating adaptive decision-making, safety assurance, and robust control. The system comprises four components: environment, decision network, safety-efficiency mechanism, and robust control. The environment updates AV states and computes rewards based on control commands. The decision network selects actions that balance safety and efficiency. The safety-efficiency mechanism includes a route planner, which assists in lane selection during merging, and an action inspector that filters unsafe actions, especially in interactions with HDVs. For control execution, MPC ensures that the chosen actions are translated into smooth and reliable commands. In emergency scenarios, the action inspector identifies approaching emergency vehicles and triggers appropriate responses, such as yielding or changing lanes. The route planner concurrently adjusts the AV's trajectory to minimize interference, ensuring compliance with safety norms.

III. KAN-ENHANCED DQN METHOD

The K-DQN network consists of a replay memory, a KAN-Q-network, and a target Q-network. The KAN-Q-network

TABLE I
VARIABLES AND DESCRIPTION

Variable	Description
s_t, a_t, r_t	State, action, and reward at time step t
$Q(s, a; \theta)$	Approximate action-value function with parameters θ
$Q^*(s, a)$	Optimal action-value function
θ, θ'	Parameters of the current Q-network and target Q-network
α_i, β_i	Learnable coefficients in the KAN activation function
λ_1, λ_2	Regularization coefficients in the KAN architecture
$\mathcal{L}(\theta)$	Loss function for training the Q-network
γ	Discount factor
$\mathbb{E}$	Expectation operator
$f(x)$	Output from the KAN layer
W, b	Weight and bias of the output layer
j	Index of the output layer neuron
n	Total number of output layer neurons
$\Phi_{l,i,j}$	Spline functions in the approximation theory
$\Phi_{l,i,j}^k$	k -th order B-spline functions in the approximation theory
C, G	Constant and Grid size in the approximation theory
L_b, L_{spline}	Lipschitz constants for activation, and spline functions
L_{Q^*}	Lipschitz constant for the optimal action-value function
ε	Approximation error
$\alpha_{\max}, \beta_{\max}$	Maximum values of α_i and β_i

processes environmental data to compute Q-values for safe, efficient decision-making, leveraging robust and precise learning. The target Q-network shares the same structure but updates less frequently to reduce learning instability. Its parameters are periodically synchronized with the KAN-Q-network. Key variables used in the K-DQN's mathematical derivations are summarized in Table I.

A. Basic DQN

DQN integrates deep learning with Q-learning to handle high-dimensional state spaces. It uses a neural network to approximate the optimal action-value function $Q^*(s, a)$ —the maximum expected return for taking action a in state s . DQN relies on experience replay and target networks.

Experience replay stores transitions (s_t, a_t, r_t, s_{t+1}) for repeated learning, enhancing stability. The target network, updated periodically from the main Q-network, computes target Q-values to reduce correlations and stabilize training.

Let $Q(s_t, a_t)$ denote the Q-value for taking action a_t in state s_t at time step t . In Q-learning, the update rule for the Q-value is given by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (1)$$

where α is the learning rate, r_t is the immediate reward, γ is the discount factor, and $\max_a Q(s_{t+1}, a)$ is the maximum Q-value over actions in the next state s_{t+1} .

Let y be the ideal (target) Q-value of the current action calculated from the Bellman equation at time step t . During the training process, y is computed by

$$y = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta') \quad (2)$$

where $\max_{a'} Q(s_{t+1}, a'; \theta')$ is the maximum Q-value over actions a' in the next state s_{t+1} , estimated with parameters θ' . Note that the subscript t denotes values at time t , while unsubscripted variables are general.

The loss function is defined on the difference between $Q(s_t, a_t; \theta)$ and y as follows:

$$\mathcal{L}(\theta) = \mathbb{E} \left[(y - Q(s_t, a_t; \theta))^2 \right] \quad (3)$$

where θ is the target network parameter, and $\mathbb{E}$ is the expectation over all state-action pairs (s_t, a_t) during training.

In the DQN framework, the goal is to minimize the loss function $\mathcal{L}(\theta)$. After computing the loss, the Q-network's weights are updated by descending the gradient to reduce $\mathcal{L}$. The gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ used for this update is given by

$$\frac{\partial \mathcal{L}}{\partial \theta} = \mathbb{E} \left[2(Q(s_t, a_t; \theta) - y) \frac{\partial Q}{\partial \theta} \right]. \quad (4)$$

Gradients are essential for updating the DQN parameters, as they measure how the loss function changes with respect to the Q-network's parameters and indicate the direction of steepest descent. By computing these gradients, we identify how to adjust the parameters to effectively minimize the loss. Specifically, we update the parameters θ using gradient descent by moving in the negative gradient direction:

$$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}}{\partial \theta} \quad (5)$$

where α is the learning rate controlling the update step size. This process repeats until the loss converges, aligning the Q-network's predictions with the targets.

Basic DQN faces challenges in complex environments like roundabouts: training instability from correlated samples and moving targets, the exploration-exploitation trade-off, sensitivity to hyperparameters, and overestimation bias causing suboptimal policies. To balance exploration and exploitation, we use a dynamic ϵ -greedy strategy, gradually reducing ϵ from 0.9 to 0.1 for broad exploration and stable refinement. KAN's adaptive spline activations enhance feature representation and reduce hyperparameter sensitivity,

improving robustness. Overestimation bias is mitigated by KAN's accurate Q-value approximation and the action inspector's real-time safety checks, preventing unsafe actions from skewing policy updates.

B. Structure of KAN

KAN's core uses spline-based activation functions of the form:

$$f(x_i; \theta_i, \beta_i, \alpha_i) = \alpha_i \cdot \text{spline}(x_i; \theta_i) + \beta_i \cdot b(x_i) \quad (6)$$

where x_i is the input to the i -th neuron, $\text{spline}(x_i; \theta_i)$ represents the spline function parameterized by coefficient θ_i , $b(x_i) = \text{SiLU}(x_i) = x_i / (1 + e^{-x_i})$ is an activation function, and α_i and β_i are learnable coefficients. Spline functions are piecewise polynomials that can approximate any continuous function. By tuning their parameters, KAN can model complex nonlinear functions.

The coefficient θ_i is updated via gradient descent on the loss $\mathcal{L}(\theta)$ in (3), following the update rule:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta_i} \quad (7)$$

with the learning rate η .

KAN uses regularization to reduce overfitting by adding to $\mathcal{L}(\theta)$ the term:

$$\mathcal{R}(\theta) = \lambda_1 \sum_i |\theta_i| + \lambda_2 \sum_i \sum_{j \neq i} |\theta_i - \theta_j| \quad (8)$$

where λ_1 and λ_2 are regularization coefficients. The L_1 term $\lambda_1 \sum_i |\theta_i|$ promotes sparsity, while $\lambda_2 \sum_i \sum_{j \neq i} |\theta_i - \theta_j|$ enforces smoothness across neurons, enhancing stability. Overall, adding $\mathcal{R}(\theta)$ controls model complexity and improves generalization.

KAN also uses parameter sharing among neurons, defined as:

$$\theta_{\text{shared}} = \frac{1}{N_{\text{group}}} \sum_{i \in \text{group}} \theta_i \quad (9)$$

where group indexes neurons sharing parameters, and N_{group} is the group size. Shared parameters θ_{shared} average neuron parameters, reducing model complexity, improving efficiency, and enhancing generalization.

These elements of the KAN architecture collectively enhance the flexibility and efficiency of the learning process, whilst ensuring robustness against overfitting and maintaining high performance across reinforcement learning tasks.

C. KAN Enhanced DQN

Integrating KAN into DQN (K-DQN) enhances Q-function approximation, boosting learning robustness and policy performance in complex DRL tasks. To justify pairing KAN with DQN over other RL methods, it's crucial to analyze their differences in handling environments and KAN's activation function traits.

To model roundabout driving as a Markov Decision Process, we define these key components:

State Space $\mathcal{S}$: It consists of the ego vehicle's position, velocity, and heading, as well as the relative positions, velocities, and headings of the surrounding vehicles within a certain range. The state at time t is represented as

$$s_t = [p_{EV}(t), v_{EV}(t), h_{EV}(t), p_{NV}^i(t), v_{NV}^i(t), h_{NV}^i(t)]^\top \quad (10)$$

where $p_{EV}(t)$, $v_{EV}(t)$, and $h_{EV}(t)$ denote the position, velocity, and heading of the ego vehicle, while $p_{NV}^i(t)$, $v_{NV}^i(t)$, and $h_{NV}^i(t)$ are the position, velocity, and heading of the i -th neighboring vehicle.

Action Space $\mathcal{A}$: It is discrete and consists of five high-level actions: faster, slower, idle, turn right, and turn left.

Reward Function r : It encourages the ego vehicle to drive safely and efficiently through the roundabout, designed as:

$$r(s_t, a_t) = w_1 r_c + w_2 r_s + w_3 r_{lc} + w_4 r_h + w_5 r_a \quad (11)$$

where r_c assigns a large penalty (-100) for collisions, r_s provides continuous feedback proportional to the vehicle's speed ($v/v_{\max}$), r_{lc} adds a small negative value (-10) for each lane change to prevent unnecessary maneuvers, r_h encourages maintaining safe distances by scaling with inverse headway time, and r_a gives a positive reward ($+200$) for reaching the target exit. The weights are empirically set as $w_1 = 1.0$, $w_2 = 0.3$, $w_3 = 0.2$, $w_4 = 0.3$, and $w_5 = 0.2$ to balance safety with efficiency.

By using (6), the goal is to directly approximate the optimal action-value function

$$\begin{aligned} Q(s, a; \theta) &= \sum_j W_j^T f(x) + b \\ &= \sum_{j \in [1-n]} \sum_{i=1}^m (\alpha_i \cdot \text{spline}(x_i; \theta_i) + \beta_i \cdot b(x_i)) + b \\ &= \sum_{j \in [1-n]} \sum_{i=1}^m (\alpha_i \cdot \text{spline}((s, a)_i; \theta_i) \\ &\quad + \beta_i \cdot b((s, a)_i)) + b \end{aligned} \quad (12)$$

where W and b are the weight and bias of the network, $f(x)$ is the output from the KAN layer, j is the index of the output layer neuron, and n is the total number of output layer neurons.

Theorem 1: (Approximation theory [29]) Suppose that a function $f(x)$ admits a representation $f = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x$, where each part Φ_l is $(k+1)$ -times continuously differentiable. Then there exist k -th order B-spline functions Φ_l^G such that for any $0 \leq m \leq k$,

$$\|f - (\Phi_{L-1}^G \circ \Phi_{L-2}^G \circ \dots \circ \Phi_1^G \circ \Phi_0^G)x\|_{C^m} \leq CG^{-k-1+m} \quad (13)$$

where C is a constant and G is the grid size. The magnitude of derivatives up to order m is measured by the C^m -norm as

$$\|g\|_{C^m} = \max_{|\beta| \leq m} \sup_{x \in [0,1]^n} |D^\beta g(x)|. \quad (14)$$

We aim to prove that under the conditions of Theorem 1, DQN with KAN as the backbone network can effectively approximate the optimal action-value function $Q^*(s, a)$. Assume the true action-value function for taking action a in state s is $Q^*(s, a)$. Our goal is to find an

approximation function $Q(s, a; \theta)$ that is as close as possible to $Q^*(s, a)$.

Considering the mean squared error properties of DQN and y given in (2), we have

$$\mathbb{E}[(Q(s_t, a_t; \theta) - Q^*(s_t, a_t))^2] = \mathbb{E}[(Q(s_t, a_t; \theta) - y)^2] + C \quad (15)$$

where $C = \mathbb{E}[(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta') - Q^*(s_t, a_t))^2]$ is a constant independent of θ . Therefore, minimizing the loss function $\mathcal{L}(\theta)$ is equivalent to minimizing the mean squared error between the approximate value function $Q(s_t, a_t; \theta)$ and the target value $r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta')$.

When we use KAN as the backbone network in DQN, the optimization objective can be rewritten as

$$\min_{\theta} \mathbb{E}[(Q(s_t, a_t; \theta) - (r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta')))^2]. \quad (16)$$

By using (12), $Q(s_t, a_t; \theta)$ can be defined as:

$$\begin{aligned} Q(s_t, a_t; \theta) &= \sum_j \sum_{i=1}^m (\alpha_i \cdot \text{spline}((s_t, a_t)_i; \theta_i) \\ &\quad + \beta_i \cdot b((s_t, a_t)_i)) + b. \end{aligned} \quad (17)$$

Since the spline functions and SiLU(x) in (6) used in KAN are continuously differentiable, the conditions of Theorem 1 are satisfied. By applying Theorem 1, we can conclude that for any state-action pair (s, a) , there exists an optimal set of parameters θ^* such that $Q(s, a; \theta^*)$ in (17) can arbitrarily approximate the optimal action-value function $Q^*(s, a)$.

Theorem 2: Let $Q(s, a; \theta)$ be the approximate action-value function defined by (12), where the spline functions $\text{spline}(x; \theta)$ and the activation function $b(x)$ are Lipschitz continuous with Lipschitz constants L_{spline} and L_b , respectively. Assume that the optimal action-value function $Q^*(s, a)$ is also Lipschitz continuous with Lipschitz constant L_{Q^*} . Then, for any $\varepsilon > 0$, there exists a set of parameters θ^* such that

$$\|Q(s, a; \theta^*) - Q^*(s, a)\|_{\infty} \leq \varepsilon, \quad (18)$$

and for any $\theta \in \Theta$,

$$\|Q(s, a; \theta) - Q^*(s, a)\|_{\infty} \leq \varepsilon + C \|\theta - \theta^*\|_2 \quad (19)$$

where $C = \sqrt{m}(\alpha_{\max} L_{\text{spline}} + \beta_{\max} L_b)$, m is the number of basis functions used in the spline approximation, $\alpha_{\max} = \max_i \alpha_i$, and $\beta_{\max} = \max_i \beta_i$.

Proof: By the universal approximation theorem for spline functions [34], $\forall \varepsilon > 0$, there is a θ^* such that

$$\|Q(s, a; \theta^*) - Q^*(s, a)\|_{\infty} \leq \varepsilon. \quad (20)$$

For any $\theta \in \Theta$, we have

$$\begin{aligned} \|Q(s, a; \theta) - Q^*(s, a)\|_{\infty} &\leq \|Q(s, a; \theta) - Q(s, a; \theta^*)\|_{\infty} \\ &\quad + \|Q(s, a; \theta^*) - Q^*(s, a)\|_{\infty} \\ &\leq \|Q(s, a; \theta) - Q(s, a; \theta^*)\|_{\infty} + \varepsilon. \end{aligned} \quad (21)$$

By the Lipschitz continuity of $\text{spline}(x; \theta)$ and $b(x)$, we obtain

$$\begin{aligned} & \|Q(s, a; \theta) - Q(s, a; \theta^*)\|_\infty \\ & \leq \sum_{j \in [1-n]} \sum_{i=1}^m (\alpha_i L_{\text{spline}} \|\theta_i - \theta_i^*\|_2 + \beta_i L_b \|\theta_i - \theta_i^*\|_2) \\ & \leq \sqrt{m}(\alpha_{\max} L_{\text{spline}} + \beta_{\max} L_b) \|\theta - \theta^*\|_2. \end{aligned} \quad (22)$$

Combining (21) and (22) gives (19). $\square$

Theorem 2 provides a quantitative bound on the approximation error between the learned action-value function $Q(s, a; \theta)$ and the optimal one $Q^*(s, a)$. The bound consists of two terms: (i) The universal approximation error ε , which can be made arbitrarily small by choosing a suitable θ^* . (ii) The term $C \|\theta - \theta^*\|_2$, which depends on the distance between the learned parameters θ and the optimal parameters θ^* . The Lipschitz continuity of the spline functions and the activation function, as well as the bound on the coefficients α_i and β_i , ensure the stability and generalization of the learned action-value function. As the training progresses and θ approaches θ^* , the approximation error decreases, indicating the convergence of the learned action-value function to the optimal one.

Under Theorem 2, by minimizing the loss function (3), DQN combined with KAN can effectively approximate the optimal action-value function $Q^*(s, a)$, as demonstrated by:

$$\lim_{\theta \rightarrow \theta^*} \mathcal{L}(\theta) \rightarrow 0 \implies \lim_{\theta \rightarrow \theta^*} Q(s, a; \theta) \rightarrow Q^*(s, a). \quad (23)$$

The optimal policy π^* selects actions maximizing the optimal Q-value Q^* for each state:

$$\pi^*(a | s) := \arg \max_a Q^*(s, a). \quad (24)$$

Thus, K-DQN can approximate $Q^*(s, a)$ by minimizing the loss function (3), enabling it to learn the optimal policy π^* . This highlights KAN's effectiveness in enhancing DQN through direct optimization and strong theoretical guarantees.

D. Computational Complexity Analysis

Integrating KAN into DQN adds computational overhead from spline-based activations. While traditional DQN's forward pass has complexity $O(LN^2)$ for L layers and N neurons, K-DQN's complexity increases to $O(LN^2 + LNS)$, where S is the number of spline segments.

During training, the backward propagation in traditional DQN has complexity $O(LN^2)$. For K-DQN, the gradient computation through spline functions adds an overhead, resulting in $O(LN^2 + LNS)$ complexity. However, two factors help mitigate this computational cost: 1) Parameter sharing among neurons reduces the number of parameters to be updated, lowering the practical computational burden to approximately $O(LN^2 + LNS)$, where $\bar{S} < S$ is the effective number of unique spline segments. 2) The improved approximation capabilities of KAN typically require fewer training iterations for convergence. Our empirical results show that K-DQN achieves equivalent performance with approximately 30% fewer training iterations compared to traditional DQN.

During inference, with fixed spline coefficients, forward pass complexity reduces to $O(LN^2 + LN)$, causing only a slight increase in decision time over traditional DQN.

IV. ROUTES PLANNER AND ACTIONS INSPECTOR

This section presents mechanisms for safe and efficient roundabout driving, covering HDV driving rules, the action inspector, and the route planner.

A. Driving Rules of HDVs

This subsection outlines HDV priority rules in roundabouts to maintain traffic flow and safety, addressing common scenarios.

1) *Entry Rule*: When an HDV nears a roundabout, it must yield to vehicles already in the entrance it plans to use, ensuring smooth traffic flow and minimizing conflicts. Such a rule is described as:

$$\text{HDV}_{\text{entering}} \not\leftarrow \text{if } \exists \text{ HDV}_{\text{passing}}. \quad (25)$$

Inside the roundabout, ego HDVs (EHDVs) must adjust their speed per the Intelligent Driver Model (IDM) policy in (26) to maintain a safe gap from the front HDV (FHDV) until exiting, preventing rear-end collisions and ensuring smooth flow.

$$a_{\text{IDM}} = a_{\max} [1 - (v_{\text{FHDV}}/v_e)^4 - (h^*/h)^2] \quad (26)$$

where $a_{\max}$ is the maximum acceleration of EHDV, v_{FHDV} is the velocity of FHDV with the desired value v_e , and h is the real gap between EHDV and FHDV. h^* is the desired gap between EHDV and FHDV with the formula

$$h^* = h_e + v_{\text{AV}} T_e - v_{\text{AV}} \Delta v / (2\sqrt{a_{\max} c}) \quad (27)$$

where h_e is the expected space to FHDV, v_{AV} is AV's speed, T_e is the desired time gap, Δv is the velocity difference between EHDV and FHDV, and c is the comfortable deceleration.

2) *Inner Lane Following Rule*: HDVs in the inner lane of the roundabout must align their speeds with the nearest vehicle ahead, even if that vehicle is in the outer lane. This rule is intended to synchronize speeds across lanes and enhance the cohesive flow of traffic, particularly in multi-lane roundabouts.

B. Route Planner

The integrated route planner for the ego vehicle (EV) comprises initial-lane selection decisions, a path-planning algorithm, and a lane-change selection mechanism. The initial-lane selection is guided by the TTC metric for each lane, ensuring safety and efficiency from the start. The path planning algorithm employs a node-based shortest path calculation to determine the most optimal route. The lane-change selection mechanism is driven by a proposed lane change cost formula, facilitating effective and strategic lane changes.

1) *Initial-Lane Selection*: By computing the TTC between the ego vehicle and surrounding vehicles, the safety levels can be ensured and unsafe actions can be avoided. In this scenario, the more potential space for driving and safety are the major considerations, thus we calculate the TTC for the inner and outer lanes as follows:

$$\begin{aligned} \text{TTC}_{\text{inner}} &= \frac{\text{Distance to HDV}_{\text{inner}}}{\text{Speed of EV} - \text{Speed of HDV}_{\text{inner}}}, \\ \text{TTC}_{\text{outer}} &= \frac{\text{Distance to HDV}_{\text{outer}}}{\text{Speed of EV} - \text{Speed of HDV}_{\text{outer}}}. \end{aligned} \quad (28)$$

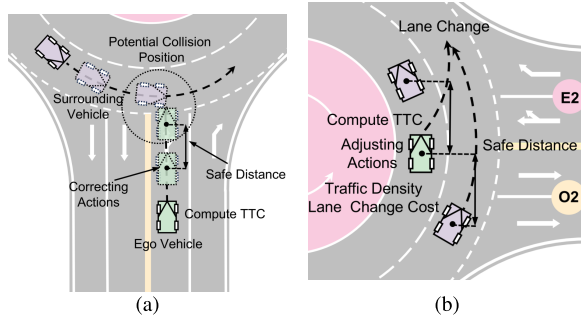


Fig. 4. Lane selection cases: (a) One HDV in the outer lane—EV chooses the safer lane with higher TTC; (b) One HDV in each lane—EV selects the lane with higher TTC.

The obtained TTC of both lanes can then be used to make the initial-lane selection rules. This paper considers several situations: No HDVs present, One HDV in outer lane, One HDV in both lanes, and Multiple HDVs in both lanes. These scenarios are described as follows:

- **No HDVs present:** The lane selection rule is

$$\text{Lane}_{\text{selected}} = \text{Inner lane} \quad \text{if HDVs} = 0. \quad (29)$$

With no HDVs present, the AV selects the inner lane for its shorter, more direct path through the roundabout.

- **One HDV in outer lane:** Fig. 4(a) illustrates this scenario, where the EV computes the TTC to maintain a safe distance from surrounding vehicles and merge into the inner lane.

- **One HDV in both lanes:** As illustrated in Fig. 4(b), by evaluating the TTC of both lanes, the lane with the higher TTC is selected for safety and more driving space. If having equal TTC values, the inner lane is chosen to enhance efficiency. The rule is summarized as

$$\text{Lane}_{\text{selected}} = \begin{cases} \text{Inner lane,} & \text{if } \text{TTC}_{\text{inner}} \geq \text{TTC}_{\text{outer}} \\ \text{Outer lane,} & \text{otherwise.} \end{cases} \quad (30)$$

If two HDVs have the same velocity but the inner-lane HDV is farther from the EV, the inner lane is selected.

- **Multiple HDVs in both lanes:** This is the most complex scenario, with two HDVs in both lanes. When multiple vehicles (more than two) are present, a weighted decision based on TTC and Total Driving Time (TDT) to the outlet is applied:

$$\text{TTC}_{\text{weighted}} = w_1 \cdot \text{TTC}_{\text{nearest}} + w_2 \cdot \text{TDT},$$

$$\text{TDT} = \sum \text{Driving time of each HDV to EV's outlet}, \quad (31)$$

where w_1 and w_2 are predefined weights reflecting traffic model preferences. The lane with the lower score is chosen to enhance safety and avoid HDV delays. The full selection process is outlined in Algorithm 1.

2) *In-Roundabout Lane Selection:* After entering the roundabout and selecting an initial lane, the next stage is path planning. We adopt a modified Breadth-First Search (BFS) [35] method that considers both distance and traffic conditions to compute the optimal path from a start point to a target within a graph structure, where nodes represent intersections in the road network, and edges represent drivable roads. The modified BFS algorithm uses the cost function:

$$C(e) = w_1 \cdot D(e) + w_2 \cdot \mathcal{D}(e) \quad (32)$$

Algorithm 1 Action Priority List for EV

Input: L : Lane index in the roundabout; $\alpha_1, \alpha_2, \alpha_3, \alpha_4$: Coefficients for priority score computation; T_h : Prediction horizon for trajectory planning; $A_{t,i}$: Feasible actions for vehicle i at time t

Output: P_t : Priority list of actions for the EV.

- 1: Determine the presence of HDVs in the roundabout.
- 2: **if** no HDVs present **then**
- 3: Select the inner lane.
- 4: **else**
- 5: Compute TTC for each HDV in both lanes.
- 6: **if** one HDV in each lane **then**
- 7: Choose lane having the highest TTC value,
- 8: preferring inner lane if equal.
- 9: **else if** multiple HDVs in both lanes **then**
- 10: Use $\text{TTC}_{\text{weighted}}$ in (31) to select lane.
- 11: **end if**
- 12: **end if**
- 13: Initialize action priority list P_t for the EV.
- 14: **for** each feasible action a_{feasible} in $A_{t,i}$ **do**
- 15: Compute the priority score of a_{feasible} .
- 16: Add a_{feasible} to P_t according to its priority score.
- 17: **end for**

where $C(e)$ is the cost of edge e , $D(e)$ is the distance of edge e , $\mathcal{D}(e)$ is the traffic density of edge e , and w_1 and w_2 are weight factors that determine the relative importance of distance and traffic density. $\mathcal{D}(e)$ is calculated by

$$\mathcal{D}(e) = N_e / L_e \quad (33)$$

where N_e is the number of vehicles on edge e , and L_e is the length of edge e . The modified BFS is formulated as

$$\text{BFS}(s, g) = \min\{p : s \rightarrow \dots \rightarrow g \mid p \in \text{Paths}(s, g)\} \quad (34)$$

where s is the start node, g is the goal node, and $\text{Paths}(s, g)$ is the set of all possible paths. The optimal path is given as

$$p^* := \arg \min_{p \in \text{Paths}(s, g)} \sum_{e \in p} C(e) \quad (35)$$

where p^* is the optimal path.

3) *Lane Selection Mechanism:* As the scenario illustrated in Fig. 4(b), traffic density ($\mathcal{D}$) and lane-change cost (C), are computed. Additionally, to align with real-world driving behaviors where vehicles preparing to exit typically move to the outer lane in advance, we implement a dynamic priority adjustment mechanism.

- **Traffic Density** $\mathcal{D}$ is calculated by iterating over all vehicles to count the number on a specified node and lane, and adjusting the density value based on vehicles' relative positions, with closer vehicles having a higher weight. When the EV is at lane node n , the density is

$$\mathcal{D}(n, l) = \sum_{NV \in N_V} \mathbf{1}_{(NV.n=n \wedge NV.l=l)} - \mathbf{1}_{(NV.n=n \wedge NV.l \neq l)} \quad (36)$$

where n indicates the node, l indicates the lane, N_V is the set of neighbor vehicles, and $\mathbf{1}$ is the indicator function:

$$\mathbf{1}_{\text{condition}} = \begin{cases} 1, & \text{if the condition is true} \\ 0, & \text{otherwise.} \end{cases} \quad (37)$$

• **Lane Change Cost** $\mathcal{C}$ is obtained by computing the distance between the controlled vehicle and other vehicles. The costs increase sharply if the distance is less than a threshold safety distance D_{safe} . The cost formula is

$$\mathcal{C}(n, l) = \sum_{NV \in N_V} \frac{D_{safe}}{D(EV, NV)} \cdot \mathbf{1}_{(D(EV, NV) < D_{safe})} \quad (38)$$

where $D(EV, NV) = \|p_{EV}(t) - p_{NV}(t)\|_2$ is the distance between EV and non-ego vehicle (NV), $p_{EV}(t)$ is EV's position, and $p_{NV}(t)$ is NV's position. This inter-vehicle distance-based cost calculation adapts naturally to different lane geometries without parameter adjustment.

• **Outer Lane Preference** $\omega(d)$ is designed to encourage timely transitions to the outer lane when approaching the target exit. This preference weight is defined as:

$$\omega(d) = \begin{cases} 0, & \text{if } d < 0.5 d_{total} \\ \beta \cdot \frac{d - 0.5 d_{total}}{0.5 d_{total}}, & \text{otherwise} \end{cases} \quad (39)$$

where d is the distance traveled from the entrance, d_{total} is the total distance to the target exit, and β is a weighting parameter (set to 0.3 in our experiments) that controls the strength of outer lane preference. The lane choice l_c is defined as:

$$l_c := \arg \min_{l \in \{0,1\}} (\mathcal{D}(n, l) + \mathcal{C}(n, l) - \omega(d) \cdot \mathbf{1}_{(l=outer)}) \quad (40)$$

This formulation ensures that as the vehicle approaches its exit (when $d > 0.5 d_{total}$), the outer lane becomes increasingly preferable, reflecting realistic driving behavior while maintaining safety through density and lane change cost terms. When lane costs are equal, the decision is refined based on the vehicle's position. This enhances both safety and efficiency by integrating real-time traffic conditions with potential lane change risks, while preserving natural driving tendencies. The route planner leverages a modified BFS algorithm with an edge selection function to identify optimal paths, providing a robust solution for autonomous navigation in roundabouts.

C. Action Inspector

Each EV plans its acceleration through the roundabout, with an incentive to accelerate for efficiency. However, to ensure safety, the EV predicts the trajectories of nearby NVs whenever their distances fall below a safety threshold D_{safe} , as shown in Fig. 4(b). This safety distance accounts for vehicle dimensions by expanding the EV's boundary by half of its width (1.05 m) and length (2.35 m), ensuring sufficient clearance for maneuvers. The vehicle-dimension-based safety calculation ensures robustness to varying lane widths. If predicted trajectories overlap, the EV switches to a following mode using the IDM policy (26), adapting to the nearest preceding vehicle and synchronizing with traffic flow. A penalty is imposed if the EV's speed remains below the expected value for more than three seconds, encouraging timely progression while maintaining safety.

1) **Safety Margin Calculation:** This margin is used to guide decision-making when selecting driving actions. As vehicles maintain a wider angle relative to each other while in proximity, the likelihood of their paths intersecting decreases.

Algorithm 2 Action Execution for EV in Roundabout

Input: P_t : Priority list of actions for the EV, initialized and populated as per the previous algorithm.

Output: $a_{t,i}$: The optimal action for vehicle i at time step t , chosen and executed from the priority list.

```

1: while  $P_t$  is not empty do
2:    $a_t \leftarrow P_t[0]$ 
3:   for  $NV$  in  $N_V$  and  $D(EV, NV) \leq D_{safe}$  do
4:     if  $EV_{a_t}$  and  $NV$  trajectories overlap in  $T_n$  then
5:       if  $NV$  in same lane and in front then
6:         Use IDM (26) to follow FHDV; break
7:       else if  $NV$  in adjacent lane then
8:         Replace  $a_t$  with the next highest priority
          action in  $P_t$ .
9:       end if
10:    end if
11:  end for
12:  if no overlap then
13:    Execute  $a_t$ 
14:  end if
15:  Remove  $a_t$  from  $P_t$ 
16: end while

```

Therefore, the safety margin for each vehicle's maneuver is defined as the minimum difference in relative angle, D_a , or the shortest time until a potential collision could occur.

$$\text{Safety Margin} = \min_{a \in A_{feasible}} D_{a,a,k} \quad (41)$$

where $A_{feasible}$ is the set of feasible actions and $D_{a,a,k}$ is the safety margin angle under action a at prediction time k .

2) **Decision-Making Criteria:** For each decision point, the AV will calculate safety margins for multiple options. If the safety margins are equivalent, the AV will prefer the lane that optimizes the route, typically the inner lane in roundabouts due to the shorter path length to the exit.

3) **Dangerous Action Replacement:** When a dangerous action is detected, the inspector replaces it with the next highest-priority action from P_t , ensuring the EV chooses the safest option. If none are safe, the EV follows the nearest vehicle using IDM until a safe action appears.

4) **Update Rule:** After each EV action, the next highest-priority action is chosen. This repeats until the EV safely exits the roundabout, with the action inspector continuously replacing risky actions with safer ones (see Algorithm 2).

The action inspector adapts by predicting T_n steps ahead and continuously monitoring trajectories. For dynamic traffic density changes, the inspector updates its safety assessments at each time step using real-time traffic information. The system quickly handles unexpected HDV behaviors using priority-based action replacement and IDM following, which adapts to sudden speed changes of preceding vehicles.

In summary, the proposed system combines route planning and action inspection for safe, efficient AV navigation in roundabouts. TTC-based lane selection ensures safe entry, the modified BFS optimizes paths using distance and traffic data, and lane changes are guided by traffic density and cost.

The action inspector monitors safety in real time, replacing risky actions to prevent collisions. By integrating global route planning with localized real-time traffic data-based lane change decisions, the proposed system demonstrates exceptional adaptability to varying traffic conditions.

V. MPC FOR ENHANCING DRL PERFORMANCE

This section introduces the robust control for AVs including the vehicle dynamic model and MPC. The MPC controller considers the vehicle dynamics, collision avoidance, and other constraints in its optimization process. It predicts the future states of the EV and surrounding vehicles using the vehicle dynamic model and the actions of neighboring vehicles predicted by the DRL agent. The combination of DRL and MPC in the proposed framework brings several benefits: it allows DRL to focus on high-level decisions while MPC manages low-level controls; MPC can correct any imperfections in DRL decisions to ensure safe and feasible actions; and MPC provides a reliable, interpretable control strategy based on clear vehicle dynamics and constraints [36].

The EV's state is updated by

$$\begin{aligned} p_{EV}(t+1) &= p_{EV}(t) + v_{EV}(t) \cdot \cos(h_{EV}(t)) \cdot \Delta t \\ v_{EV}(t+1) &= v_{EV}(t) + u(t) \cdot \Delta t \\ h_{EV}(t+1) &= h_{EV}(t) + v_{EV}(t) \cdot \tan(\delta(t)) \cdot \Delta t / L \end{aligned} \quad (42)$$

where Δt is the sampling time, $v_{AV}(t)$ is the speed, $h_{AV}(t)$ is the heading angle, L is the wheelbase length, $u(t)$ is the acceleration, and $\delta(t)$ is the steering angle.

The following control input and collision avoidance constraints are applied to ensure safety and feasibility:

$$\begin{aligned} v_{\min} \leq v_{EV}(t) \leq v_{\max}, \quad a_{\min} \leq u(t) \leq a_{\max}, \\ \delta_{\min} \leq \delta(t) \leq \delta_{\max}, \quad \|p_{EV}(t) - p_{NV}(t)\|_2 \geq D_{safe}. \end{aligned} \quad (43)$$

At time step t , the optimal solutions $u^*(t)$ and $\delta^*(t)$ are obtained by solving the optimization problem:

$$\begin{aligned} \min J_c \\ \text{s.t. (42), (43), } NV \in N_V, k \in [0, N_p - 1] \end{aligned} \quad (44)$$

with the cost function $J_c = \sum_{k=0}^{N_p-1} (v_{AV}(k) - v_{AV}^*)^2 + \sum_{k=0}^{N_p-1} \sum_{i=1}^{N_v} (\|p_{SV}^i(k) - p_{AV}^i(k)\| - D_{safe})^2 + \lambda \sum_{k=0}^{N_c-1} u^2(k)$. N_p and N_c represent the prediction horizon and control horizon, respectively. $v_{AV}^*(k)$ is the target speed, and λ is a given weighting factor. In our experiments, we set N_p as 10 and the N_c as 5. The entire control process is summarized in Algorithm 3.

VI. SIMULATION RESULTS

We evaluate K-DQN in the roundabout scenario described in Section II, focusing on training efficiency and collision rate. All experiments are conducted using three random seeds, with mean results plotted and standard deviations shown as shaded areas. The roundabout follows standard traffic engineering design, with an inner radius of 20 m, outer radius of 28 m, and a lane width of 4 m. All vehicles are modeled as typical passenger cars (4.7 m long, 2.1 m wide), and these dimensions

Algorithm 3 MPC Controller for Adjusting EV's Velocity

Input: v_{EV}^* : The target speed of the ego vehicle.

Output: $u[0]$: The optimal control input for the first time step, or the output of the PID controller if no solution is found.

```

1: MPC_Controller  $v_{EV}^*$ 
2:  $EV \leftarrow \text{deepcopy}(\text{self})$ 
3:  $N_V \leftarrow \text{get\_surrounding\_vehicles}()$ 
4:  $opti \leftarrow \text{ca.Opti}()$ 
5:  $u \leftarrow opti.variable(N)$ 
6:  $J_c \leftarrow 0$ 
7: for  $k \leftarrow 0$  to  $N - 1$  do
8:   for vehicle in  $N_V$  do
9:     action  $\leftarrow \text{use\_K-DQN}$ 
10:     $\_to\_predict\_vehicle\_action(\text{vehicle})$ 
11:   end for
12:    $\delta(k) \leftarrow \text{compute\_steering}(EV, NV)$ 
13:    $EV.update(u(k), \delta(k))$ 
14:    $J_c \leftarrow J_c + (v_{EV}(k) - v_{EV}^*)^2$ 
15:   for  $i \leftarrow 1$  to  $N_v$  do
16:      $J_c \leftarrow J_c + (\|p_{NV}^i(k) - p_{EV}^i(k)\|_2 - D_{safe})^2$ 
17:   end for
18:    $J_c \leftarrow J_c + \lambda u^2(k)$ 
19:    $\text{add\_vehicle\_constraints}(EV, NV, u(k))$ 
20: end for
21:  $opti.minimize(J_c)$ 
22:  $solution \leftarrow opti.solve()$ 
23: if  $solution$  found then
24:   return  $solution.value(u[0])$ 
25: else
26:   return  $\text{PID\_controller}(v_{EV}(0), v_{EV}^*)$ 
27: end if

```

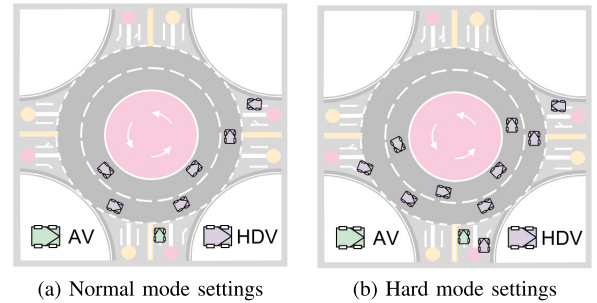


Fig. 5. Validation settings: (a) Normal mode with six initial HDVs; (b) Hard mode with ten HDVs.

are considered in collision detection and safety distance calculations. The algorithm employs a relative state representation and distance-based safety mechanism, ensuring robustness to varying lane widths. Vehicles exiting the roundabout leave the AV's observation range but continue updating their kinematics. We examine three scenarios as follows:

- *Ablation study in hard mode:* The proposed K-DQN is compared with K-DQN without the action inspector, K-DQN without MPC, and the baseline DQN.
- *Normal mode validation:* The proposed K-DQN is compared with benchmarks with seven initial vehicles in the roundabout as depicted in Fig. 5(a).

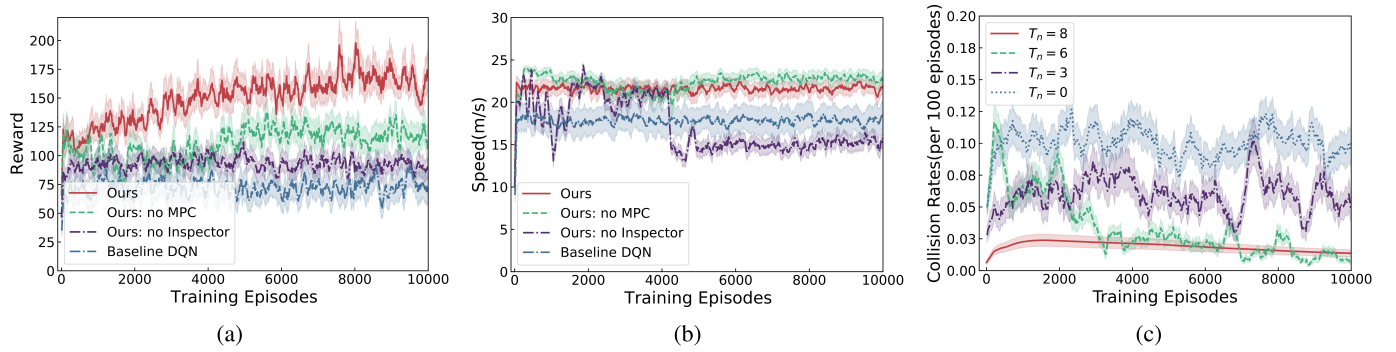


Fig. 6. Performance comparison with different components and prediction steps. (a) reward, (b) speed, and (c) collision rates under different prediction steps (T_n). The shaded regions denote the standard deviations over 3 random seeds.

- *Hard mode validation:* The proposed K-DQN is compared with benchmarks with eleven initial vehicles in the roundabout as depicted in Fig. 5(b).

The benchmarks used for comparison in the normal and hard mode validations include PPO [23], A2C [37], ACKTR [38], and DQN [28]. The performance metrics used for evaluation include training convergence rate, collision rate, average speed, and reward values during training and evaluation. Considering the inherent risks associated with real-world vehicles and the constraints imposed by legal regulations, scenario-based virtual testing offers significant benefits like precise environmental replication and enhanced testing efficiency. Therefore, this study employs scenarios developed on the Highway virtual simulation platform [39]. To ensure robust performance under varying traffic conditions, HDV speeds are initialized following a normal distribution around a mean speed of 20 m/s with a $\pm 15\%$ variation. Similarly, HDV's positions are randomized with a normal distribution to create diverse traffic patterns. At the end of each episode, the vehicles and their velocities are slightly randomized at their spawn points to enhance the generalization capability of our proposed model. This randomization helps evaluate the model's adaptability to different traffic speeds and densities, which is crucial for real-world deployment. The computer and environment setup for this study include Python 3.6, PyTorch 1.10.0, Ubuntu 20.04.6 LTS OS, a Intel® Core™ i5-12600KF CPU, an NVIDIA GeForce RTX 3090 GPU, and 64GB of RAM.

A. Ablation Study in Hard Mode

This section describes experiments to evaluate the crucial functions of the action inspector and MPC of the proposed system in hard mode. To comprehensively assess the contribution of each component, we include a baseline DQN (using traditional MLP architecture) for comparison with our K-DQN variants. We divide the experiments into four validations: Validation 1 evaluates training performance of K-DQN, K-DQN without action inspector, K-DQN without MPC, and baseline DQN. Validation 2 tests the stability of the speed variations. Validation 3 compares the reward across the evaluation. Validation 4 analyzes the number of collisions.

Validation 1: Training performance. To better assess performance under random HDV behavior, we conducted tests

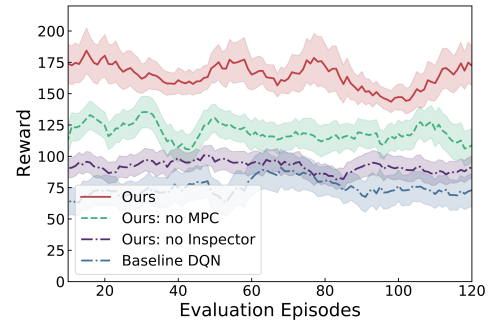


Fig. 7. Rewards of different K-DQN schemes and DQN.

using three random seeds and varied scenarios. Figure 6(a) compares training curves of the proposed K-DQN with three variants: K-DQN without action inspector, K-DQN without MPC, and baseline DQN with a standard MLP architecture. As expected, the full K-DQN consistently achieves higher peak rewards and faster, more stable convergence. Compared to baseline DQN, the K-DQN without inspector already benefits from the KAN architecture, yielding 10–15% higher rewards. However, the largest gain comes from the action inspector, which filters unsafe actions and reduces collision penalties during exploration. The proposed K-DQN shows low variance across seeds, indicating stable training, while K-DQN without the inspector or MPC suffers from more fluctuations and slower convergence. The baseline DQN performs the worst, confirming that both the KAN architecture and action inspector are essential for robust and efficient learning.

Validation 2: Stability of the speeds variation. Figure 6(b) compares the speed profiles across different configurations: the proposed K-DQN, K-DQN without action inspector, K-DQN without MPC, and baseline DQN. The baseline DQN maintains relatively stable yet suboptimal speeds around 17–18 m/s, reflecting a conservative strategy due to ϵ -greedy exploration without safety constraints, which often leads to aggressive or overly cautious decisions in dense traffic [24]. K-DQN without inspector exhibits significant early fluctuations (episodes 0–4000), as the KAN architecture aggressively explores without safety filtering, eventually stabilizing at a lower speed of about 15 m/s to mitigate collision risk. This shows that while KAN improves representation, it cannot alone resolve the safety-efficiency trade-off. In contrast, the full

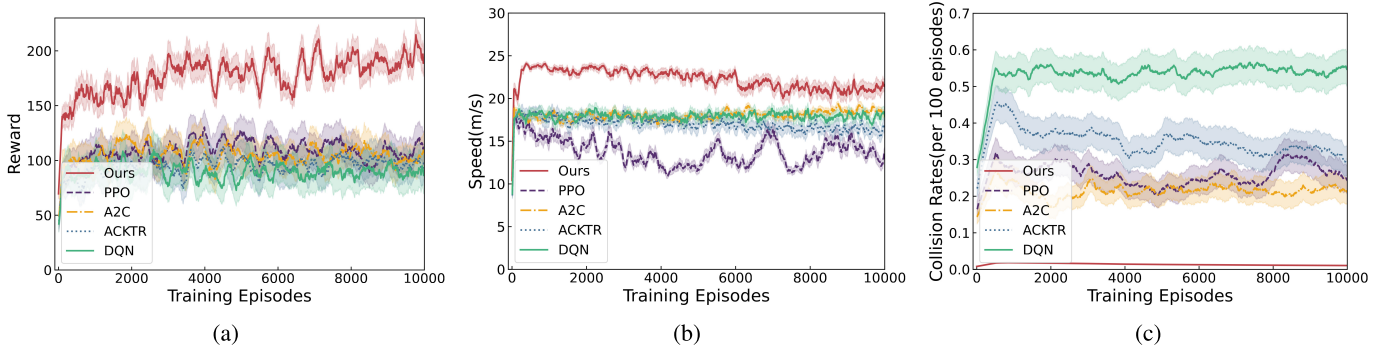


Fig. 8. Performance comparison with benchmarks in normal mode: (a) reward, (b) speed, and (c) collision rate.

TABLE II
COLLISION RATES AND AVERAGE SPEEDS FOR
DIFFERENT K-DQN SCHEMES

Metrics	No MPC	No Inspector	Ours
Collision Rate (%)	9	11	2
Average Speed (m/s)	22.88	16.23	22.37

Collision rate is measured per 100 episodes during training. The best results are highlighted in bold.

K-DQN achieves the best balance, maintaining higher speeds near 22 m/s with minimal variation (< 2 m/s), while K-DQN without MPC shows larger fluctuations around 4 m/s. These results highlight the complementary roles of the action inspector, which enables safe high-speed exploration, and MPC, which ensures stable policy execution. This does not undermine the value of DQN as a foundation; rather, it underscores the effectiveness of our enhancements. The KAN architecture enhances function approximation for safer policy learning, the action inspector reduces collisions dramatically (from 52% to 2%) by filtering unsafe actions, and MPC guarantees smooth, reliable execution. Together, these components enable DQN-based systems to reach state-of-the-art performance in safety-critical autonomous driving scenarios.

Figure 6(c) shows how prediction horizon (T_n) affects the action inspector. With $T_n = 8$, the collision rate is lowest and most stable (~ 0.02), allowing ample time for action replacement. Shorter horizons ($T_n = 6$ and 3) raise the rate to ~ 0.03 and ~ 0.07 , while $T_n = 0$ performs worst (~ 0.11), highlighting the importance of predictive action inspection.

Validation 3: Reward values among evaluation. Figure 7 compares rewards for K-DQN, its ablated variants, and baseline DQN. K-DQN achieves the highest average (~ 175), showing the benefit of integrating the action inspector and MPC. Removing MPC drops rewards to ~ 125 , and without the inspector, rewards fall below 100 due to frequent collisions. Lastly, the baseline DQN performs the worst, with rewards mostly staying below 75, underscoring the limitations of naive reinforcement learning without structured safety or strategic reasoning. These results collectively highlight the necessity of combining multi-level safety and planning mechanisms for robust autonomous decision-making.

Validation 4: Collision rate. Table II compares collisions and average speed for the proposed K-DQN, K-DQN without

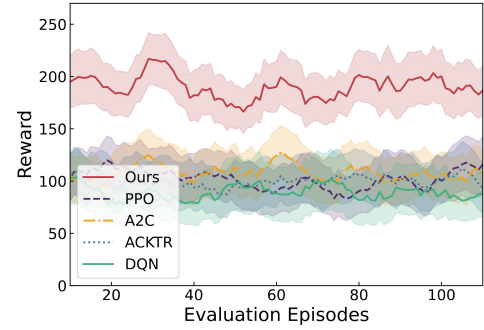


Fig. 9. Rewards for K-DQN and benchmarks: normal mode.

the action inspector, and K-DQN without MPC. The full K-DQN achieves the lowest collision rate at 2% with a solid average speed of 16.23 m/s. Without the action inspector, speed drops and collisions rise to 11%. Removing MPC increases speed to 22.88 m/s, but the collision rate remains high at 9%. These results highlight the full K-DQN's strong balance of safety and efficiency.

B. Normal Mode Validation

This section presents the experiments in normal mode (with seven initial vehicles in the roundabout in Fig. 5(a)) with comparison to benchmark DRL algorithms, PPO [23], A2C [37], ACKTR [38], and DQN [28].

Validation 1: Training performance. To assess training performance, we test K-DQN and benchmarks using three random seeds and varied scenarios. Figure 8(a) shows K-DQN achieves the highest rewards (over 200) and fastest convergence, outperforming PPO (~ 125), A2C and ACKTR (similar), and DQN (~ 80). Figure 8(b) shows K-DQN maintains the highest and most stable speed (22 m/s), while PPO has the lowest and most unstable. A2C reaches 18 m/s; others are similar. Figure 8(c) shows K-DQN achieves the lowest collision rate (< 0.05), compared to PPO and A2C (~ 0.2), ACKTR (~ 0.35), and DQN (~ 0.55).

Validation 2: Reward values among evaluation. Figure 9 illustrates the reward comparison between the proposed K-DQN and other benchmark algorithms. DQN has the lowest reward during the evaluation, falling below 75. A2C and ACKTR are similar, both increasing the reward to around 75. PPO has a relatively higher reward of around 100,

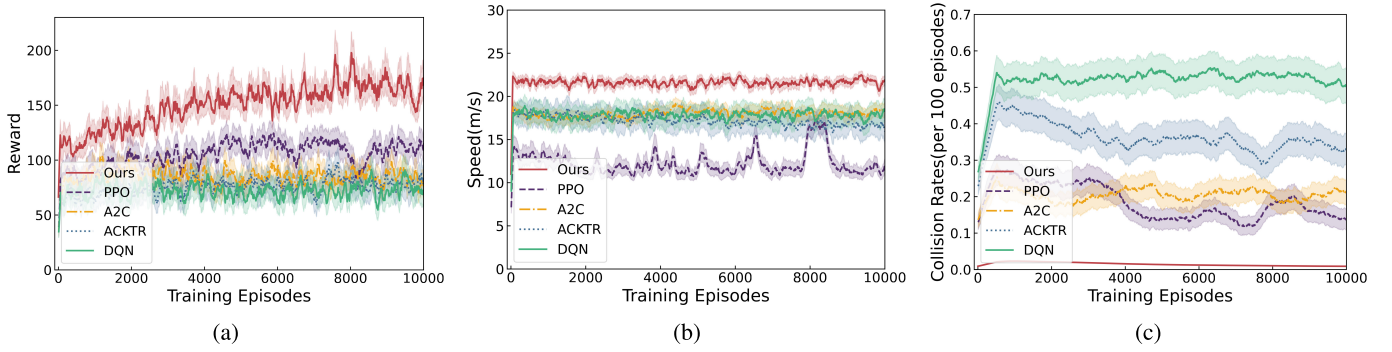


Fig. 10. Performance comparison with benchmarks in hard mode: (a) reward, (b) speed, and (c) collision rate.

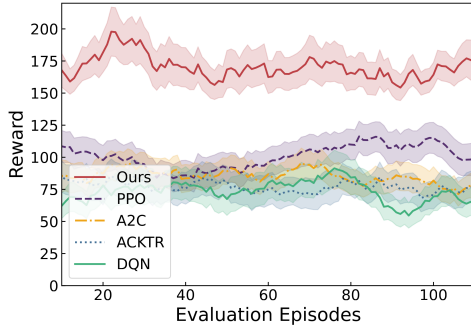


Fig. 11. Rewards for K-DQN and benchmarks: hard mode.

peaking at 125. The reward of the proposed K-DQN fluctuates around 175, significantly surpassing other benchmark algorithms.

C. Hard Mode Validation

This validation assesses the proposed system's safety and efficiency in the most challenging conditions, with eleven initial vehicles in the roundabout as depicted in Fig. 5(b).

Validation 1: Training performance. To ensure a robust evaluation, three random seeds are used to generate diverse scenarios for comparison with benchmark algorithms. Figure 10(a) shows K-DQN achieved the highest peak rewards (~ 160) and the fastest convergence, outperforming the closest competitor (~ 100) and significantly surpassing DQN (75). In Fig. 10(b), K-DQN maintains higher and more stable speeds, peaking at 2 m/s, while PPO shows the lowest and most unstable performance (11–16 m/s). A2C, ACKTR, and DQN follows similar trends, with A2C reaching up to 18 m/s. Figure 10(c) highlights K-DQN's superior safety, with a collision rate below 0.05, compared to DQN (0.52), ACKTR (0.35), and the more variable rates of A2C (~ 0.2) and PPO (~ 0.15). These results confirm K-DQN's strong training efficiency, stability, and safety, supporting its potential for real-world deployment.

Validation 2: Reward values among evaluation. The evaluation, using three random seeds to ensure scenario diversity (Fig. 11), highlights the K-DQN algorithm's superior reward performance. Traditional DQN remained below 70, while A2C and ACKTR show slight improvements near 75. PPO reaches a peak of 125 but averaged around 100. In contrast, K-DQN

TABLE III
COLLISION RATES AND AVERAGE SPEEDS FOR THE PROPOSED METHOD AND BENCHMARKS

Scenarios	Metrics	PPO	A2C	ACKTR	DQN	Ours
Normal Mode	coll. rate (%)	23	21	28	52	1
	avg. v (m/s)	14.76	18.83	17.89	18.31	21.59
Hard Mode	coll. rate (%)	12	19	31	52	2
	avg. v (m/s)	13.67	18.04	17.53	17.70	22.52

coll. rate means collision rate per 100 episodes during training. The best results are highlighted in bold.

consistently outperforms all baselines, maintaining reward levels around 175, demonstrating clear dominance in reward maximization.

Table III compares collision rates and average speeds of the proposed K-DQN against PPO, A2C, ACKTR, and DQN in both normal and hard modes. In normal mode, K-DQN achieves the lowest collision rate (0.01), outperforming PPO (0.23), A2C (0.21), ACKTR (0.28), and DQN (0.52). It also records the highest average speed at 21.59 m/s, surpassing PPO (14.76 m/s), A2C (18.83 m/s), ACKTR (17.89 m/s), and DQN (18.31 m/s). In hard mode, K-DQN maintains its advantage with the lowest collision rate (0.02) and highest speed (22.52 m/s), while all benchmarks show reduced safety and efficiency. These results confirm K-DQN's superior safety, efficiency, and robustness across varying traffic complexities.

This study employs average vehicle speed as a key performance metric, reflecting roundabout capacity and traffic efficiency. Higher average speeds imply improved flow and increased capacity. Comparative analysis shows that the proposed K-DQN consistently outperforms benchmark algorithms in both safety and efficiency. While PPO exhibits moderate performance, it lags behind K-DQN in both metrics. A2C and ACKTR perform better than PPO and DQN but fall short of K-DQN. DQN records the highest collision rate, despite maintaining relatively high speeds. Overall, K-DQN achieves lower collision rates and higher average speeds across both normal and hard scenarios, highlighting its effectiveness for safe, efficient navigation in complex traffic environments.

VII. CONCLUSION

This paper proposes a DRL-based algorithm to improve AV safety and efficiency in complex roundabout traffic with HDVs. Using a DQN that processes surrounding vehicle states,

it avoids manual feature engineering and enhances environmental perception. The integration of a KAN further improves learning accuracy and reliability. The algorithm includes an action inspector to reduce collisions, a route planner for efficient driving, and MPC control for stable, precise actions. Evaluations show superior performance with fewer collisions, reduced travel times, and faster training convergence compared to state-of-the-art benchmarks. Future research will focus on: 1) evaluating the algorithm's robustness in more complex traffic scenarios, including urban ramps with elevation changes, multi-lane intersections with pedestrian crossings, and roundabouts with varied lane widths (3.5–5.0 m) and radii to assess adaptability to diverse infrastructures; 2) enhancing driving strategies by incorporating passenger comfort (jerk minimization) and energy efficiency for electric vehicles; and 3) extending the algorithm to collaborative multi-agent settings with V2V communication and platooning to evaluate performance in cooperative and competitive urban traffic.

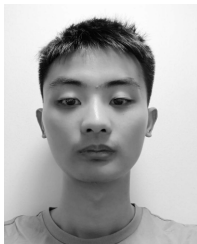
REFERENCES

- [1] P. Ladosz, M. Mammadov, H. Shin, W. Shin, and H. Oh, "Autonomous landing on a moving platform using vision-based deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 9, no. 5, pp. 4575–4582, May 2024.
- [2] S. Hou, C. Wang, and J. Gao, "Reinforced stable matching for crowd-sourced delivery systems under stochastic driver acceptance behavior," *Transp. Res. C, Emerg. Technol.*, vol. 170, Jan. 2025, Art. no. 104916.
- [3] X. Ma and X. He, "Providing real-time en-route suggestions to CAVs for congestion mitigation: A two-way deep reinforcement learning approach," *Transp. Res. B, Methodol.*, vol. 189, Nov. 2024, Art. no. 103014.
- [4] J. Wang and L. Sun, "Robust dynamic bus control: A distributional multi-agent reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4075–4088, Apr. 2023.
- [5] W. Wu, Y. Zhu, and R. Liu, "Dynamic scheduling of flexible bus services with hybrid requests and fairness: Heuristics-guided multi-agent reinforcement learning with imitation learning," *Transp. Res. B, Methodol.*, vol. 190, Dec. 2024, Art. no. 103069.
- [6] C. Badué et al., "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Aug. 2020, Art. no. 113816.
- [7] S. Mak, L. Xu, T. Pearce, M. Ostroumov, and A. Brintrup, "Fair collaborative vehicle routing: A deep multi-agent reinforcement learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 157, Dec. 2023, Art. no. 104376.
- [8] X. Xing, Z. Zhou, Y. Li, B. Xiao, and Y. Xun, "Multi-UAV adaptive cooperative formation trajectory planning based on an improved MATD3 algorithm of deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12484–12499, Sep. 2024.
- [9] PatentPC. (2023). *Autonomous Vehicle Market Growth in 2020–2030: 50+ Key Stats You Need to Know*. Accessed: May 25, 2025. [Online]. Available: <https://patentpc.com/blog/autonomous-vehicle-market-growth-in-2020-2030-50-key-stats-you-need-to-know>
- [10] J. Yu, P.-A. Laharotte, Y. Han, W. Ma, and L. Leclercq, "Perimeter control with heterogeneous metering rates for cordon signals: A physics-regularized multi-agent reinforcement learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 171, Feb. 2025, Art. no. 104944.
- [11] J. Xi, F. Zhu, P. Ye, Y. Lv, G. Xiong, and F.-Y. Wang, "Auxiliary network enhanced hierarchical graph reinforcement learning for vehicle repositioning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11563–11575, Sep. 2024.
- [12] Q. Ma, X. Wang, S. Zhang, and C. Lu, "Distributed self-organizing control of CAVs between multiple adjacent-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5430–5441, May 2023.
- [13] D. Chen, Q. Qi, Q. Fu, J. Wang, J. Liao, and Z. Han, "Transformer-based reinforcement learning for scalable multi-UAV area coverage," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 10062–10077, Aug. 2024.
- [14] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, "An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14458–14469, Dec. 2020.
- [15] X. Wang, S. Zhang, and H. Peng, "Comprehensive safety evaluation of highly automated vehicles at the roundabout scenario," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20873–20888, Nov. 2022.
- [16] J. F. Medina-Lee, J. Godoy, A. Artuñedo, and J. Villagra, "Speed profile generation strategy for efficient merging of automated vehicles on roundabouts with realistic traffic," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 3, pp. 2448–2462, Mar. 2023.
- [17] F. Mao, Z. Li, and L. Li, "A comparison of deep reinforcement learning models for isolated traffic signal control," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 1, pp. 160–180, Jan. 2023.
- [18] E. Debada, A. Ung, and D. Gillet, "Occlusion-aware motion planning at roundabouts," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 276–287, Jun. 2021.
- [19] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. Girard, "Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2211–2226, Mar. 2022.
- [20] Y. Shi, Z. Gu, X. Yang, Y. Li, and Z. Chu, "An adaptive route guidance model considering the effect of traffic signals based on deep reinforcement learning," *IEEE Intell. Transp. Syst. Mag.*, vol. 16, no. 3, pp. 21–34, May 2024.
- [21] K. Cai, Z. Li, T. Guo, and W. Du, "Multi-airport departure scheduling via multiagent reinforcement learning," *IEEE Intell. Transp. Syst. Mag.*, vol. 16, no. 2, pp. 102–116, Mar. 2024.
- [22] H. Lei, H. Ran, I. S. Ansari, K.-H. Park, G. Pan, and M.-S. Alouini, "DDPG-based aerial secure data collection," *IEEE Trans. Commun.*, vol. 72, no. 8, pp. 5179–5193, Aug. 2024.
- [23] B. Peng, Y. Xie, G. Seco-Granados, H. Wymeersch, and E. A. Jorswieck, "Communication scheduling by deep reinforcement learning for remote traffic state estimation with Bayesian inference," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4287–4300, Apr. 2022.
- [24] R. Li, W. Gong, L. Wang, C. Lu, Z. Pan, and X. Zhuang, "Double DQN-based coevolution for green distributed heterogeneous hybrid flowshop scheduling with multiple priorities of jobs," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 4, pp. 6550–6562, Oct. 2024.
- [25] G. Basile, A. Petrillo, and S. Santini, "DDPG based end-to-end driving enhanced with safe anomaly detection functionality for autonomous vehicles," in *Proc. IEEE Int. Conf. Metrol. Extended Reality, Artif. Intell. Neural Eng. (MetroXR/INE)*, Oct. 2022, pp. 248–253.
- [26] Z. Liu, Y. Cao, J. Chen, and J. Li, "A hierarchical reinforcement learning algorithm based on attention mechanism for UAV autonomous navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13309–13320, Nov. 2023.
- [27] X. Liu, M. Yu, C. Yang, L. Zhou, H. Wang, and H. Zhou, "Value distribution DDPG with dual-prioritized experience replay for coordinated control of coal-fired power generation systems," *IEEE Trans. Ind. Informat.*, vol. 20, no. 6, pp. 8181–8194, Jun. 2024.
- [28] P. Cai, H. Wang, Y. Sun, and M. Liu, "DQ-GAT: Towards safe and efficient autonomous driving with deep Q-learning and graph attention networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21102–21112, Nov. 2022.
- [29] Z. Liu et al., "KAN: Kolmogorov-arnold networks," 2024, *arXiv:2404.19756*.
- [30] A. Kundu, A. Sarkar, and A. Sadhu, "KANQAS: Kolmogorov–Arnold network for quantum architecture search," *EPJ Quantum Technol.*, vol. 11, no. 1, p. 76, Nov. 2024.
- [31] S. Mallick, F. Airaldi, A. Dabiri, and B. De Schutter, "Multi-agent reinforcement learning via distributed MPC as a function approximator," *Automatica*, vol. 167, Sep. 2024, Art. no. 111803.
- [32] Y. Bie, Y. Ji, and D. Ma, "Multi-agent deep reinforcement learning collaborative traffic signal control method considering intersection heterogeneity," *Transp. Res. C, Emerg. Technol.*, vol. 164, Jul. 2024, Art. no. 104663.
- [33] I. Kempf, P. J. Goulart, and S. R. Duncan, "Control of cross-directional systems with approximate symmetries," *Automatica*, vol. 167, Sep. 2024, Art. no. 111782.
- [34] G. Pillonetto and M. Bisiacco, "Kernel-based linear system identification: When does the representer theorem hold?" *Automatica*, vol. 159, Jan. 2024, Art. no. 111347.
- [35] T. Chen, Y. Zhang, X. Qian, and J. Li, "A knowledge graph-based method for epidemic contact tracing in public transportation," *Transp. Res. C, Emerg. Technol.*, vol. 137, Apr. 2022, Art. no. 103587.

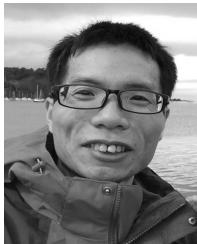
- [36] Y. Yuan, S. Li, L. Yang, and Z. Gao, "Nonlinear model predictive control to automatic train regulation of metro system: An exact solution for embedded applications," *Automatica*, vol. 162, Apr. 2024, Art. no. 111533.
- [37] Y. Hou, G. Han, F. Zhang, C. Lin, J. Peng, and L. Liu, "Distributional soft actor-critic-based multi-AUV cooperative pursuit for maritime security protection," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 6049–6060, Jun. 2024.
- [38] Y. Zhu, Z. Wang, Y. Zhu, C. Chen, and D. Zhao, "Discretizing continuous action space with unimodal probability distributions for on-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 6, pp. 1–13, Jun. 2025.
- [39] E. Leurent. (2018). *An Environment for Autonomous Driving Decision-Making*. [Online]. Available: <https://github.com/eleurent/highway-env>



Zhihao Lin received the M.S. degree from the College of Electronic Science and Engineering, Jilin University, Jilin, China. He is currently pursuing the Ph.D. degree with the College of Science and Engineering, University of Glasgow, Glasgow, U.K. His research interests include multi-sensor fusion SLAM systems, reinforcement learning, and hybrid control of vehicle platoons.



Zhen Tian received the B.S. degree in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K. He is currently pursuing the Ph.D. degree with the College of Science and Engineering, University of Glasgow, Glasgow. His research interests include interactive vehicle decision systems and autonomous racing decision systems.



Jianglin Lan received the Ph.D. degree from the University of Hull in 2017. He has been a Leverhulme Early Career Fellow and a Lecturer at the University of Glasgow since 2022. He was a Visiting Professor at the Robotics Institute, Carnegie Mellon University, in 2023. From 2017 to 2022, he held post-doctoral positions at Imperial College London, Loughborough University, and The University of Sheffield. His research interests include safe AI, fault-tolerant systems, autonomous vehicles, and robotics.



Qi Zhang received the B.S. degree from North University of China, Taiyuan, China, in 2022, and the M.S. degree from the School of Computing Science, University of Glasgow, Scotland, U.K. He is currently pursuing the Ph.D. degree with the University of Amsterdam, The Netherlands. His current research interests include algorithms and systems for semantic SLAM in dynamic environments.



Ziyang Ye received the B.S. degree in computer science from The University of Adelaide, Adelaide, SA, Australia, in 2020, where he is currently pursuing the M.S. degree in artificial intelligence and machine learning. His research interests include 2D/3D computer vision tasks and reinforcement learning.



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



Xianxian Zhao received the Ph.D. degree in electrical engineering from the University of Birmingham in 2018. She is currently the Principle Investigator of the SEAI-RD&D Program at University College Dublin. Her research interests include modeling and control of electric machines and converters, stability and power quality of power-electronic-based power systems, and model order reduction of large power systems.