KAN-LSTM Enhanced Multi-Agent Advantage Actor-Critic Reinforcement Learning for Autonomous Ramp Merging

Zhihao Lin, Jianglin Lan, and Xianxian Zhao, *Member, IEEE*

***Abstract*—Autonomous ramp merging represents one of the most challenging scenarios for autonomous vehicles (AVs), requiring safe interaction with human-driven vehicles (HDVs) while maintaining traffic efficiency. This paper proposes a novel KL-MA2C framework, which integrates Kolmogorov-Arnold Networks (KAN) and Long Short-Term Memory (LSTM) into the Multi-Agent Advantage Actor-Critic (MA2C) algorithm. The proposed framework addresses key limitations in existing approaches to robust safety mechanisms. Our key contribution lies in the synergistic integration of KAN-LSTM within a multi-agent reinforcement learning framework. KAN provides superior function approximation through adaptive spline-based activation functions, while LSTM captures temporal dependencies in human driving patterns. This combination is enhanced by an action inspector that enforces safety constraints and Model Predictive Control (MPC) that ensures smooth trajectory execution. Extensive simulations demonstrate significant safety and efficiency performance improvements over state-of-the-art methods. These results establish KL-MA2C as an effective solution for autonomous ramp merging, with potential implications for safe and efficient autonomous driving in complex multi-agent environments.**

***Index Terms*—Autonomous driving, Kolmogorov-Arnold network, long short-term memory network, model predictive control, reinforcement learning.**

I. Introduction

AUTONOMOUS vehicles (AVs) face significant challenges when navigating ramps, particularly in their interactions with human-driven vehicles (HDVs) and in determining optimal merging strategies [1]. The complexity of ramp merging stems from three fundamental sources: unpredictable human driving behaviors that create diverse trajectory patterns [2], uncertain future states of surrounding vehicles [3], and the need for real-time decision-making under both static and dynamic safety constraints as illustrated in Fig. 1. These factors necessitate robust and adaptive control mechanisms that can ensure safe and efficient interactions between AVs and HDVs while maintaining traffic flow [4], [5].

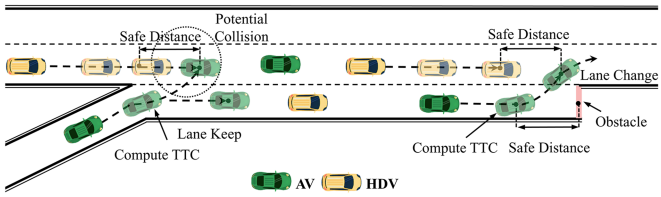


Fig. 1. Illustration of safety-critical scenarios during ramp merging. The diagram shows key interactions between AVs and HDVs, including safe distance maintenance, collision detection, and lane-changing decisions.

To address these challenges, existing approaches can be broadly categorized into behavior modeling methods and decision-making frameworks [6]. Traditional behavior modeling approaches, including Hidden Markov Models [7] and Gaussian Mixture Models [8], have demonstrated capabilities in capturing driving patterns through probabilistic state modeling. The Intelligent Driver Model (IDM) has proven effective for longitudinal car-following dynamics [9], successfully modeling acceleration patterns and collision avoidance behaviors [10]. However, these traditional methods face limitations in handling complex temporal dependencies and adapting to the dynamic nature of mixed traffic environments [11].

Long Short-Term Memory (LSTM) networks have emerged as a superior solution for temporal behavior modeling [12], addressing key limitations of traditional approaches through their ability to process long sequences of trajectory data and capture complex temporal dependencies [13]. Their memory mechanism enables effective prediction of vehicle intentions and future states across diverse driving scenarios [14], [15], making them particularly suitable for modeling human driving behavior in autonomous systems [16]. However, the integration of LSTM-based behavior modeling with advanced decision-making frameworks is less underexplored, particularly in safety-critical multi-agent scenarios.

For decision-making and control, researchers have explored various strategies ranging from traditional Model Predictive Control (MPC) [17], [18] to game-theoretic approaches [19]. While MPC excels at handling vehicle dynamics and safety constraints through receding horizon optimization [20], it

Received 19 February 2025; revised 27 May 2025 and 3 July 2025; accepted 23 July 2025. Date of publication 8 August 2025; date of current version 19 January 2026. This work was supported in part by the China Scholarship Council Ph.D. Scholarship for 2023-2027 under Grant 202206170011, in part by Leverhulme Trust Early Career Fellowship under Grant ECF-2021-517, and in part by the Sustainable Energy Authority of Ireland through RD&D Award under Grant 22/RDD/776. The review of this article was coordinated by Dr. Zheng Chen. *(Corresponding author: Xianxian Zhao.)*

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

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

Digital Object Identifier 10.1109/TVT.2025.3593661

often struggles with real-time computation and modeling uncertainties. Game-theoretic models provide insights into multi-agent interactions but typically assume unrealistic complete information scenarios [21].

Deep reinforcement learning (DRL) has emerged as a promising paradigm for autonomous driving decision-making [22]. Methods like Deep Deterministic Policy Gradient (DDPG) [23], Proximal Policy Optimization (PPO) [24], Deep Q-Network (DQN) [25], Actor Critic using Kronecker-Factored Trust Region (ACKTR) [26], and Advantage Actor-Critic (A2C) [27] have demonstrated effectiveness in various driving scenarios [28], each with distinct advantages in continuous control, safety assurance, sample efficiency, and exploration strategies respectively [29]. However, these approaches face challenges in function approximation accuracy and adaptation to complex multi-agent environments where both safety and efficiency must be optimized simultaneously.

In parallel, DRL has been widely applied to task offloading and resource management in edge and vehicular networks, which face similar challenges such as dynamic environments, multi-agent interactions, and real-time decision-making. DRL variants like Rainbow DQN [30], Asynchronous Advantage Actor-Critic (A3C) [31], [32], and Double DQN (DDQN) [33], [34] have shown effectiveness in mobile edge computing, vehicular edge computing, and blockchain-enabled internet of vehicles systems. These studies underscore the value of advanced DRL architectures in managing heterogeneous network conditions, mobility, and multi-objective optimization—challenges also central to autonomous driving in mixed traffic scenarios [35].

Recently, researchers have introduced the Kolmogorov-Arnold Network (KAN) architecture by replacing linear layers with adaptive B-spline functions [36], resulting in superior function approximation capabilities compared to traditional multilayer perceptrons. To address the identified limitations, this paper proposes KL-MA2C, a novel framework that synergistically combines KAN's enhanced representational capacity with LSTM's temporal modeling capabilities within a Multi-Agent Advantage Actor-Critic (MA2C) architecture for autonomous ramp merging. The main contributions are:

- A KAN-enhanced, LSTM-enhanced MA2C (KL-MA2C) algorithm is developed for the AV to interact with surrounding vehicles. The KAN network enables learning of safe and efficient control strategies, while the LSTM network improves the prediction of surrounding vehicle motions, assisting in safer decision-making.
- An action inspector is proposed to reduce conflicts with surrounding HDVs by imposing safety rules, significantly reducing collisions under complex traffic conditions.
- A MPC module is integrated to generate smooth and safe control commands, while a Proportional-Integral-Derivative (PID)-based fallback mechanism enhances computational robustness under potential planning failures in ramp traffic scenarios.

The remainder of the paper is organized as follows: Section II introduces the problem formulation and system model. Sections III–V detail the KL-MA2C method, action inspector, and MPC design. Section VI analyzes computational complexity, Section VII presents experimental results, and Section V) concludes with future research directions.

II. Problem Statement and System Structure

This paper studies the ramp merging scenarios (see Fig. 1) with two main lanes, a ramp lane, a sloped buffer road, and an obstacle positioned at the ramp's end. AVs initiate from the buffer road and must make two key decisions: merging onto the ramp lane and subsequently integrating into the main lanes alongside dense HDVs. To ensure safety and efficiency, AVs are tasked with timely lane changes, obstacle avoidance, and collision prevention. These are achieved through Time-To-Collision (TTC) calculations, safe following distance enforcement, and collision zone monitoring. We adopt a centralized training with decentralized execution framework to address the multi-agent nature of this problem: AVs share observations during training to learn coordinated strategies, while operating independently at execution time using only local observations. Since HDVs do not explicitly communicate their intentions, their behaviors are inferred through sensor-based temporal modeling. Safe and efficient merging thus relies on robust coordination among AVs and adaptive responses to diverse HDV driving patterns, supported by reliable control mechanisms.

To address these challenges, we propose a comprehensive system (see Fig. 2) that integrates three key components: an enhanced learning framework (detailed in Section III), a safety-focused action inspector (detailed in Section IV), and a robust control mechanism (detailed in Section V). The learning framework uses the KAN-enhanced MA2C method to generate effective coordinating control strategies for AVs, while the LSTM network improves prediction of human driver intentions and behaviors. The action inspector implements safety rules to manage interactions between AVs and HDVs, while MPC ensures smooth and safe execution of the planned actions. In practical operation, the system follows a hierarchical control structure: KL-MA2C generates high-level driving decisions based on the current traffic states and LSTM-predicted behaviors, the action inspector validates these decisions against safety constraints using TTC analysis, and MPC translates the approved actions into smooth control commands with PID fallback for computational robustness.

The proposed KL-MA2C adopts the following reward function to promote safe and efficient merging behavior in multi-agent highway scenarios:

$$r(s_t, a_t) = w_1 r_{\text{collision}} + w_2 r_{\text{speed}} + w_3 r_{\text{merging}} + w_4 r_{\text{headway}} \tag{1}$$

where $r_{\text{collision}}$ imposes a penalty for vehicle collisions to ensure safety prioritization, r_{speed} provides normalized speed rewards scaled within $[10, 30]$ m/s to encourage traffic efficiency, r_{merging} applies an exponential penalty for prolonged occupation of merging lanes to facilitate timely lane changes, and r_{headway} penalizes unsafe following distances using logarithmic headway time scaling. The reward components are weighted as $w_1 = -20$, $w_2 = 1$, $w_3 = 4$, and $w_4 = 4$. The cooperative

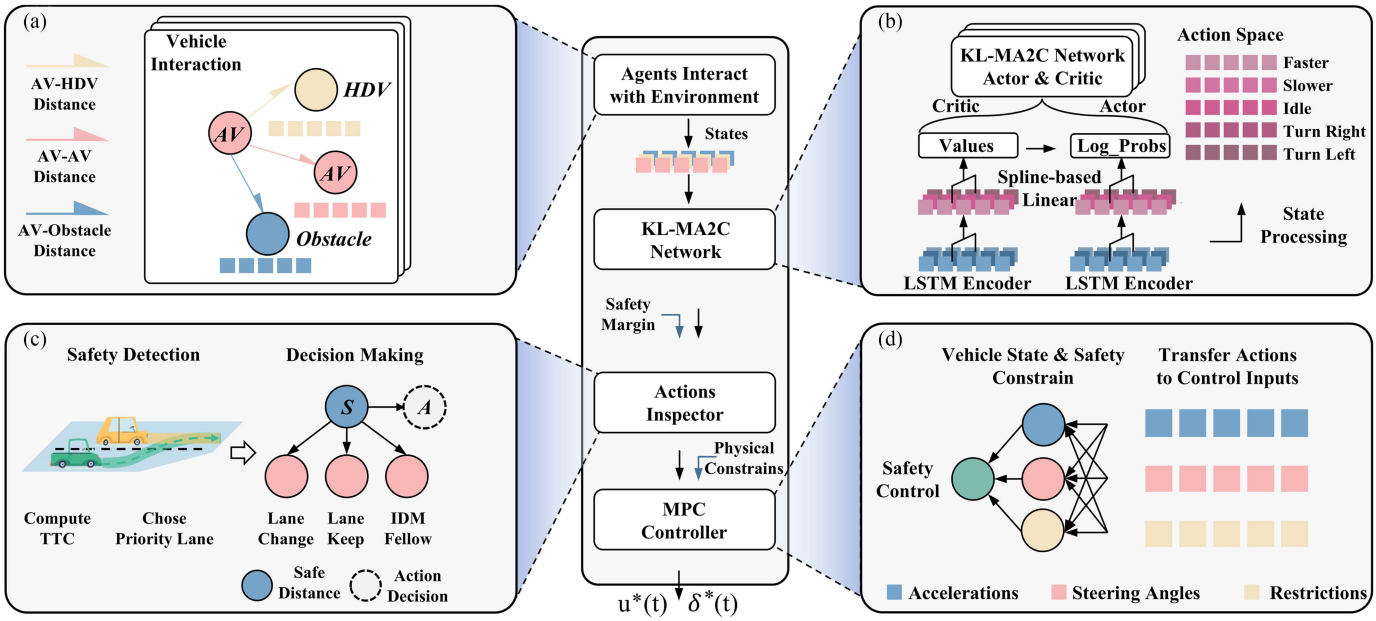


Fig. 2. The KAN-based, conflict-avoidance, and proper-lane-detection DRL system. The integrated framework operates through four key components: (a) Vehicle interaction monitoring for state observation; (b) KL-MA2C network with KAN-enhanced function approximation and LSTM temporal modeling for action-based decision-making through discrete action selection; (c) Action inspector performing safety detection through TTC computation and priority-based decision selection; and (d) MPC controller converting high-level actions to safe control inputs with physical constraint enforcement.

multi-agent reward is computed as the sum of individual agent rewards divided by the number of controlled vehicles, enabling coordinated learning across all agents.

III. KAN-Enhanced MA2C Method

A. Basic MA2C

The MA2C algorithm extends A2C to multi-agent settings, with each agent using separate actor and critic networks to learn independently from local observations and rewards.

1) Actor Network: The actor network learns a policy $\pi_\theta(a|s)$, parameterized by θ, that maps the state s to a probability distribution over actions a. The probability of taking action a in state s is given by

$$\pi_\theta(a|s) = \frac{\exp(A(s,a))}{\sum_{a' \in A} \exp(A(s,a'))} \tag{2}$$

where $A(s,a)$ represents the advantage function, which measures the relative value of taking action a in state s. The denominator sums over all possible actions a' in the action space A, ensuring that the probabilities sum to 1.

The policy gradient update for the actor is based on the advantage function defined as

$$A(s,a) = R_t - V_\phi(s) \tag{3}$$

where R_t is the actual reward obtained at time step t.

The loss function for the actor network is given by

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_{s,a\sim\pi}\left[A(s,a)\log \pi_\theta(a|s)\right]. \tag{4}$$

This loss function encourages the actor to increase the probability of actions with higher advantage values.

To encourage exploration in the policy, the actor loss is added with the entropy term

$$\mathcal{L}_{\text{entropy}} = -\lambda \sum_a \pi_\theta(a|s) \cdot \log \pi_\theta(a|s) \tag{5}$$

where λ is a regularization parameter that controls the strength of the entropy. This term encourages the policy to maintain a diverse set of actions, preventing it from becoming too deterministic and promoting exploration of the action space.

2) Critic Network: The critic network is used to estimate the value function $V_\phi(s)$ that represents the expected return from state s. $V_\phi(s)$ is parameterized by ϕ as follows:

$$V_\phi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right] \tag{6}$$

where γ is the discount factor determining the importance of future rewards, and r_t is the reward received at time step t. The expectation $\mathbb{E}_\pi$ is taken over the policy π, reflecting the average return from state s when following the policy.

The critic network is updated by minimizing the loss

$$\mathcal{L}_{\text{critic}} = 0.5(R_t - V_\phi(s))^2. \tag{7}$$

This guides the critic network to produce value estimates that closely approximate the actual returns.

B. KAN-Enhanced MA2C

The core of the KAN architecture consists in using spline-based activation functions for individual neurons. For the i-th neuron, the KAN activation function takes the form:

$$f(x_i;\theta_i,\beta_i,\alpha_i) = \alpha_i \cdot \text{spline}(x_i;\theta_i) + \beta_i \cdot \psi(x_i) \tag{8}$$

where x_i is the input to the neuron, $\text{spline}(x_i;\theta_i)$ is the B-spline function parameterized by coefficient θ_i, $\psi(x_i) = \text{SiLU}(x_i) = x_i/(1+e^{-x_i})$ is the SiLU activation function, and α_i and β_i are learnable coefficients that control the contribution of the spline and activation components, respectively.

When applying KAN to the actor and critic networks in MA2C, we extend (8) to accommodate multiple neurons within a network layer, as follows:

$$F(x;\theta,\beta,\alpha,b) = \sum_{i=1}^{m}(\alpha_i \cdot \text{spline}(x_i;\theta_i) + \beta_i \cdot \psi(x_i)) + b \tag{9}$$

where b is a learnable bias term, and m is the number of neurons in the layer. This extension enables the KAN architecture to process multi-dimensional inputs and outputs required for multi-agent reinforcement learning, where each layer must handle complex state-action spaces.

1) KAN-Enhanced Actor Network: The actor network in KAN-enhanced MA2C employs the extended KAN formulation to parameterize the policy distribution, replacing traditional linear layers with adaptive B-spline functions. The probability of selecting action a in state s is computed by

$$\pi_\theta(a|s) = \frac{\exp(f_{\text{actor}}(s,a))}{\sum_{a'\in A}\exp(f_{\text{actor}}(s,a'))} \tag{10}$$

with the actor network's output $f_{\text{actor}}(s,a)$ given by

$$f_{\text{actor}}(s,a) = \sum_{j\in[1,n]} F((s,a);\theta_j,\beta_j,\alpha_j,b_j) \tag{11}$$

where n denotes the number of network layers, and each layer's output is computed using the extended KAN structure F in (9). Note that index j represents different network layers, while index i in (9) represents neurons within each layer.

2) KAN-Enhanced Critic Network: The critic utilizes the KAN structure to approximate the value function $V_\phi(s)$ via

$$V_\phi(s) = f_{\text{critic}}(s) = \sum_{j\in[1,n]} F(s;\phi_j,\beta_j,\alpha_j,b_j) \tag{12}$$

where ϕ_j contains the learnable parameters for the j-th layer.

3) Approximation Theory for KAN-Enhanced MA2C: To demonstrate the effectiveness of KAN in enhancing MA2C, we present Lemma 1 and Theorem 1.

Lemma 1 [36]: For a function $f(x)$ represented by

$$f = (\Phi_{L-1}\circ\Phi_{L-2}\circ\cdots\circ\Phi_1\circ\Phi_0)x \tag{13}$$

where each element $\Phi_{l,i,j}$ is $(k+1)$-times continuously differentiable, there exist k-th order B-spline functions $\Phi^G_{l,i,j}$ such that for any $0\le m\le k$,

$$\|f - (\Phi^G_{L-1}\circ\Phi^G_{L-2}\circ\cdots\circ\Phi^G_1\circ\Phi^G_0)x\|_{C^m} \le CG^{-k-1+m} \tag{14}$$

where C is a constant, G is the grid size, and C^m-norm is used to measure derivatives up to order m.

By applying Lemma 1 to our framework, we reach

$$\|\pi_\theta(a|s) - \pi^*(a|s)\|_{C^m} \le C_1 G^{-k-1+m}$$

$$\|V_\phi(s) - V^*(s)\|_{C^m} \le C_2 G^{-k-1+m} \tag{15}$$

where $\pi^*(a|s)$ and $V^*(s)$ are the optimal policy and value functions, respectively.

Theorem 1: Let $\pi_\theta(a|s)$ and $V_\phi(s)$ be the approximate policy and value functions defined by KAN-enhanced MA2C in (11) and (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 policy function $\pi^*(a|s)$ and the optimal value function $V^*(s)$ are also Lipschitz continuous with Lipschitz constants L_{π^*} and L_{V^*}, respectively. Then, for any $\varepsilon_1,\varepsilon_2>0$, there exist a set of parameters θ^* and ϕ^* such that

$$\|\pi_{\theta^*}(a|s) - \pi^*(a|s)\|_\infty \le \varepsilon_1, \|V_{\phi^*}(s) - V^*(s)\|_\infty \le \varepsilon_2 \tag{16}$$

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

$$\|\pi_\theta(a|s) - \pi^*(a|s)\|_\infty \le \varepsilon_1 + C_1\|\theta-\theta^*\|_2$$
$$\|V_\phi(s) - V^*(s)\|_\infty \le \varepsilon_2 + C_2\|\phi-\phi^*\|_2 \tag{17}$$

where $C_1 = C_2 = \sqrt{m}(\alpha_{\max}L_{\text{spline}} + \beta_{\max}L_b)$, $\alpha_{\max} = \max_i \alpha_i$, and $\beta_{\max} = \max_i \beta_i$.

Proof: By using Lemma 1, for any $\varepsilon_1,\varepsilon_2>0$, there exist parameters θ^* and ϕ^* such that

$$\|\pi_{\theta^*}(a|s) - \pi^*(a|s)\|_\infty \le \varepsilon_1, \|V_{\phi^*}(s) - V^*(s)\|_\infty \le \varepsilon_2. \tag{18}$$

For any $\theta\in\Theta$ and $\phi\in\Phi$, we have

$$\|\pi_\theta(a|s) - \pi^*(a|s)\|_\infty \le \|\pi_\theta(a|s) - \pi_{\theta^*}(a|s)\|_\infty + \|\pi_{\theta^*}(a|s) - \pi^*(a|s)\|_\infty \le \|\pi_\theta(a|s) - \pi_{\theta^*}(a|s)\|_\infty + \varepsilon_1. \tag{19}$$

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

$$\begin{aligned}&\|\pi_\theta(a|s) - \pi_{\theta^*}(a|s)\|_\infty \\ &\le \sum_{j\in[1-n]}\sum_{i=1}^{m}(\alpha_i L_{\text{spline}}\|\theta_i - \theta_i^*\|_2 + \beta_i L_b\|\theta_i - \theta_i^*\|_2) \\ &\le \sqrt{m}(\alpha_{\max}L_{\text{spline}} + \beta_{\max}L_b)\|\theta-\theta^*\|_2.\end{aligned} \tag{20}$$

Therefore,

$$\|\pi_\theta(a|s) - \pi^*(a|s)\|_\infty \le \varepsilon_1 + C_1\|\theta-\theta^*\|_2 \tag{21}$$

where $C_1 = \sqrt{m}(\alpha_{\max}L_{\text{spline}} + \beta_{\max}L_b)$. The proof for the value function follows similarly. ■

1) Convergence of KAN-Enhanced MA2C: By minimizing the actor and critic loss functions, KAN-enhanced MA2C can effectively approximate the optimal policy and value functions:

$$\lim_{\theta\to\theta^*}\mathcal{L}_{\text{actor}}(\theta)\to 0 \Rightarrow \lim_{\theta\to\theta^*}\pi_\theta(a|s)\to\pi^*(a|s),$$
$$\lim_{\phi\to\phi^*}\mathcal{L}_{\text{critic}}(\phi)\to 0 \Rightarrow \lim_{\phi\to\phi^*}V_\phi(s)\to V^*(s). \tag{22}$$

This demonstrates the effectiveness of KAN in enhancing MA2C, enabling better learning and optimization of policies in multi-agent environments. The KAN network reduces parameters using a compact spline-based representation with few learnable coefficients (α_i,β_i), making its assumptions more suitable for complex multi-agent scenarios and easing training and generalization challenges in high-dimensional DRL settings.

C. KAN-LSTM Enhanced MA2C (KL-MA2C)

Building on KAN-enhanced MA2C, we integrate an LSTM network to capture temporal dependencies, forming the KAN-LSTM MA2C (KL-MA2C) framework, which combines KAN's expressiveness with LSTM's memory capabilities.

1) LSTM Cell: The LSTM cell processes sequential data using gates to regulate information flow. At each time step t, the LSTM takes as input the current state s_t and the previous hidden state h_{t-1}. The core equations of the LSTM cell are

$$f_t = \sigma(W_f \cdot [h_{t-1}, s_t] + b_f),\ i_t = \sigma(W_i \cdot [h_{t-1}, s_t] + b_i),$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, s_t] + b_C)\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t,$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, s_t] + b_o),\ h_t = o_t \odot \tanh(C_t),$$

where σ and $\tanh$ are the sigmoid and hyperbolic tangent functions, and $\odot$ denotes element-wise multiplication.

The LSTM encoder is trained independently to learn temporal representations without interfering with policy learning. Its objective function focuses on temporal consistency:

$$\mathcal{L}_{\text{LSTM}} = \mathbb{E}_{s_t, h_t, s_{t+1}} \left[\text{MSE}(f_{\text{pred}}(h_t), s_{t+1})\right] \tag{23}$$

where $f_{\text{pred}}(\cdot)$ is a prediction head that maps LSTM hidden states to future state predictions, encouraging the encoder to capture meaningful temporal patterns.

During training, the LSTM encodes current states s_t and agent identifiers into temporal features h_t. These features are concatenated with original states to form enhanced representations $\tilde{s}_t = [s_t, \text{detach}(h_t)]$ for the actor-critic networks, where the detach operation prevents gradient backpropagation from decision networks to the LSTM encoder. This decoupled approach naturally accommodates dynamic agent populations common in traffic scenarios, as each agent maintains independent temporal histories without requiring complex batching strategies. It also enables component-specific hyperparameter tuning, with the LSTM and policy networks trained at different frequencies to match their respective learning dynamics.

2) KAN-LSTM Enhanced Actor Network: The actor network in KL-MA2C combines the LSTM output with the KAN structure to generate the policy

$$\pi_\theta(a|\tilde{s}_t) = \frac{\exp(f_{\text{actor}}(\tilde{s}_t, a))}{\sum_{a' \in A} \exp(f_{\text{actor}}(\tilde{s}_t, a'))} \tag{24}$$

with $f_{\text{actor}}(\tilde{s}_t, a)$ defined using the extended KAN:

$$f_{\text{actor}}(\tilde{s}_t, a) = \sum_{j \in [1,n]} F((\tilde{s}_t, a); \theta_j, \beta_j, \alpha_j, b_j) \tag{25}$$

where F is the extended KAN formulation in (9).

3) KAN-LSTM Enhanced Critic Network: The critic network uses the KAN-LSTM to approximate the value function

$$V_\phi(\tilde{s}_t) = f_{\text{critic}}(\tilde{s}_t) = \sum_{j \in [1,n]} F(\tilde{s}_t; \phi_j, \beta_j, \alpha_j, b_j). \tag{26}$$

4) KL-MA2C Training Objective: The KL-MA2C uses a decoupled optimization strategy to tackle the challenges of multi-agent reinforcement learning with temporal dependencies. Instead of jointly optimizing all network components, it separates temporal modeling from decision-making to reduce gradient conflicts from competing objectives.

Algorithm 1: KL-MA2C Training Algorithm.

Input: N agents, state dimension d_s, LSTM hidden dimension d_h, hyperparameters $\lambda, \alpha, \gamma, \eta$
Output: Trained KAN-LSTM Actor-Critic networks

procedure Train ($\mathcal{B}$)
  Extract batch: $s, a, r, s' \leftarrow \mathcal{B}.s, \mathcal{B}.a, \mathcal{B}.r, \mathcal{B}.s'$
  Process LSTM: $h \leftarrow \text{LSTM}(\text{reshape}(s))$
  Process LSTM: $h' \leftarrow \text{LSTM}(\text{reshape}(s'))$
  Construct enhanced states: $\tilde{s} \leftarrow [s, \text{detach}(h)]$
  Construct enhanced next states: $\tilde{s}' \leftarrow [s', \text{detach}(h')]$
  LSTM training: $\mathcal{L}_{\text{LSTM}} \leftarrow \text{TrainLSTMSeparately}()$
  for agent $i \leftarrow 0$ **to** $N-1$ **do**
    Compute policy: $\pi_\theta \leftarrow \text{ActorKAN}(\tilde{s}[i])$
    Compute value: $V_\phi \leftarrow \text{CriticKAN}(\tilde{s}[i])$
    Compute next value: $V'_\phi \leftarrow \text{CriticKAN}(\tilde{s}'[i])$
    Calculate TD target: $y[i] \leftarrow r[i] + \gamma V'_\phi$
    Calculate advantage: $A[i] \leftarrow y[i] - V_\phi$
    Compute losses:
    $\mathcal{L}_{\text{actor}} \leftarrow -\log \pi_\theta(a[i]|\tilde{s}[i]) \cdot A[i]$
    $\mathcal{L}_{\text{critic}} \leftarrow \frac{1}{2}(y[i] - V_\phi)^2$
    $\mathcal{L}_{\text{entropy}} \leftarrow -\lambda \sum_a \pi_\theta(a|\tilde{s}[i]) \log \pi_\theta(a|\tilde{s}[i])$
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{actor}} + \mathcal{L}_{\text{critic}} + \mathcal{L}_{\text{entropy}}$
    Apply gradient update with learning rate η
  end for
end procedure
procedure TrainLSTMSeparately
  $\mathcal{L}_{\text{LSTM}} \leftarrow \mathbb{E}[\text{MSE}(f_{\text{pred}}(h_t), s_{t+1})]$ ▷ (23)
  Update LSTM parameters independently
  return $\mathcal{L}_{\text{LSTM}}$
end procedure
procedure ActorKAN($\tilde{s}$)
  $x \leftarrow \tilde{s}$
  for $j \leftarrow 1$ **to** n **do**
    $x \leftarrow F(x; \theta_j, \beta_j, \alpha_j, b_j)$ ▷ (25)
  end for
  return softmax(x)
end procedure
procedure CriticKAN($\tilde{s}$)
  $x \leftarrow \tilde{s}$
  for $j \leftarrow 1$ **to** n **do**
    $x \leftarrow F(x; \phi_j, \beta_j, \alpha_j, b_j)$ ▷ (26)
  end for
  return x
end procedure

The actor-critic networks are optimized using the loss:

$$\mathcal{L}_{\text{AC}} = \mathcal{L}_{\text{entropy}} + \mathcal{L}_{\text{actor}} + \mathcal{L}_{\text{critic}} \tag{27}$$

with components $\mathcal{L}_{\text{entropy}} = -\lambda \mathbb{E}_{\tilde{s}_t}[\sum_a \pi_\theta(a|\tilde{s}_t) \log \pi_\theta(a|\tilde{s}_t)]$, $\mathcal{L}_{\text{actor}} = -\mathbb{E}_{\tilde{s}_t, a \sim \pi}[A(\tilde{s}_t, a) \log \pi_\theta(a|\tilde{s}_t)]$, and $\mathcal{L}_{\text{critic}} = 0.5\mathbb{E}_{\tilde{s}_t}[A(\tilde{s}_t, a)^2]$, with advantage function $A(\tilde{s}_t, a) = R_t - V_\phi(\tilde{s}_t)$, discounted return R_t, and regularization parameter λ.

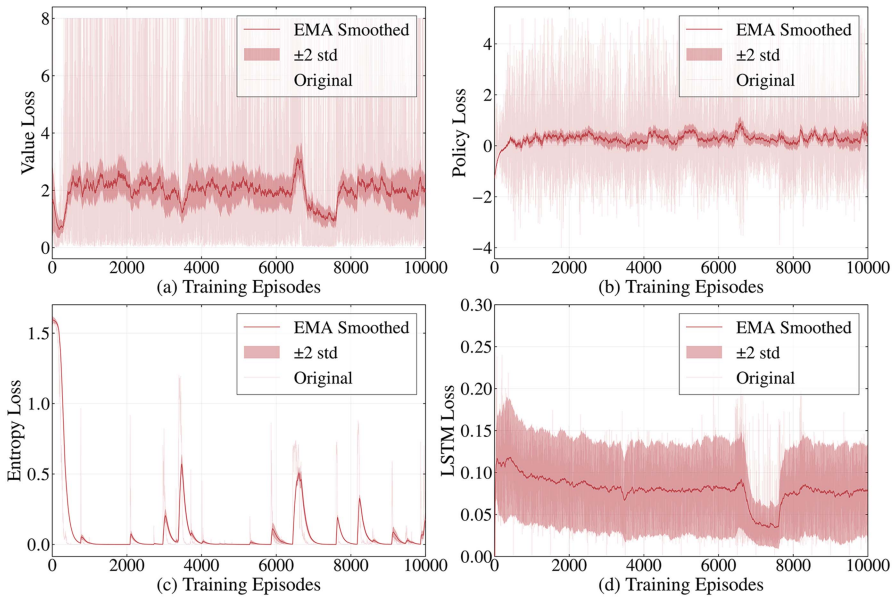


Fig. 3. Training loss trajectories of subnetworks. (a) Value loss, (b) Policy loss, (c) Entropy loss, and (d) LSTM loss.

Algorithm 1 summarizes the complete KL-MA2C training procedure. To validate the algorithm's convergence robustness, Fig. 3 presents the training loss trajectories for all subnetworks over 10,000 episodes. The results highlight four key points: 1) The value loss (Fig. 3(a)) converges smoothly to around 2.0 with minimal oscillations, demonstrating effective value function approximation despite KAN's enhanced capacity. 2) The policy loss (Fig. 3(b)) stabilizes near 0 with tight confidence intervals, confirming stable policy optimization without reward variance-induced instabilities. 3) The entropy loss (Fig. 3(c)) shows controlled periodic variations corresponding to exploration phases, while trending downward overall. 4) The LSTM loss (Fig. 3(d)) rapidly converges and remains stable below 0.1, validating efficacy of our decoupled temporal learning. Overall, these smooth convergence patterns and narrow confidence intervals ($\pm 2\sigma$) demonstrate that KL-MA2C successfully mitigates oscillation issues associated with high-capacity networks. The decoupled optimization effectively isolates temporal modeling from policy learning, preventing gradient conflicts that could destabilize training.

IV. Action Inspector

This section introduces the mechanisms to ensure safety and efficiency during interactive driving in the ramp, starting with the HDVs' driving rules and then the action inspector design.

A. Driving Rules of HDVs

1) Entry Rule: As an HDV nears the ramp entry, it must keep a safe distance from the vehicle ahead—HDV or AV—to ensure smooth flow and avoid entry conflicts. In this study, HDVs stay on the main lane and do not enter the ramp. This rule is summarized as:

$$\text{HDV}_{\text{entering}} \not\leftarrow \text{if } \exists\, \text{HDV}_{\text{passing}} \tag{28}$$

2) Car-Following Behavior: HDV car-following behavior is modeled using the IDM, which captures human driving patterns through the following acceleration rule:

$$a_{\text{IDM}} = a_{\max}[1 - (v_{\text{Front\_HDV}}/v_e)^4 - (h^*/h)^2] \tag{29}$$

where $a_{\max}$ is the maximum acceleration, $v_{\text{Front\_HDV}}$ the leading vehicle's speed, v_e the preferred speed, h the actual gap, and $h^* = h_e + v_{\text{AV}}T_e + v_{\text{AV}}\Delta v/(2\sqrt{a_{\max}c})$ the desired gap, with the standstill distance h_e, the time headway T_e, the speed difference Δv, and the comfortable deceleration c.

3) Lane-Changing Rule: HDVs on the main lane adjust speed to match the nearest vehicle ahead and may change lanes if the speed gap exceeds a threshold. With two main lanes (upper and lower), the target is the front area of the opposite lane. Lane changes occur probabilistically to reflect real behavior. The lane-changing decision is modeled as

$$\text{LaneChange} = \begin{cases} \text{True} & \text{if } v_{\text{HDV}} - v_{\text{Front\_HDV}} > \Delta v_{\text{a}} \\ \text{False} & \text{otherwise} \end{cases} \tag{30}$$

where v_{HDV} is the speed of the considered HDV, and Δv_{a} is the allowed following speed difference.

B. Action Inspector

The AV plans its acceleration through the ramp lane, favoring acceleration while assessing collision risk. It considers two lane-changing types to merge onto the main lane (Fig. 4): active, when entering from the buffer road (right subplot), and passive, when nearing the ramp's end (left subplot). In both cases, the AV predicts nearby vehicle (NV) trajectories if their distance is below the safe threshold D_{safe}, and ensures this distance is also maintained from any end-of-ramp obstacle. The AV-NV distance is computed by

$$D(\text{AV}, \text{NV}) = \|p_{\text{AV}}(t) - p_{\text{NV}}(t)\|_2 \tag{31}$$

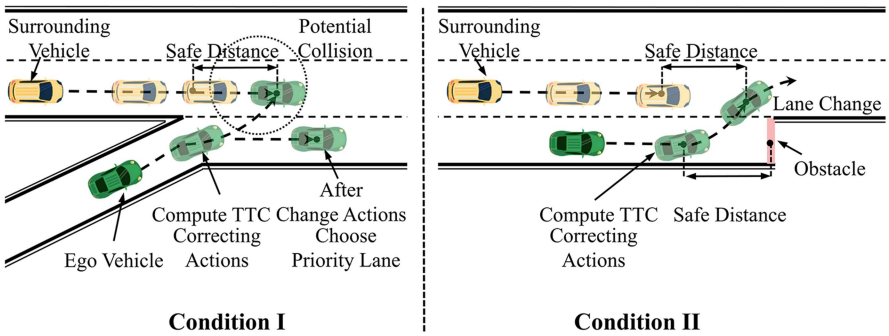


Fig. 4. Examples of Initial lane-selection (Condition I) and in-ramp lane-selection (Condition II).

Algorithm 2: Decision-Making and Update Rule for AV Navigation With TTC Detection.

Input: AV, NVs, A_{feasible}, D_{safe}, TTC_{safe}
Output: AV safely navigates to the main lane
Initialize A_{feasible}
while AV not in main lane **do**
  $\text{SM} \leftarrow \min_{a \in A_{\text{feasible}}} D_{\text{d},a,k},\ \forall a \in A_{\text{feasible}}$ ▷ Safety Margin
  $a^* \leftarrow \arg\max_{a \in A_{\text{feasible}}} \text{SM}$
  if $\exists a_1, a_2 \in A_{\text{feasible}} : \text{SM}_{a_1} = \text{SM}_{a_2}$ **then**
    $a^* \leftarrow \arg\min_{a \in \{a_1, a_2\}} t_{\text{lane-change}}(a)$
  end if
  Execute a^*
  Update A_{feasible}
  for $v \in$ NVs **do**
    if $D(\text{AV}, v) < D_{\text{safe}}$ **then**
      $a^* \leftarrow \text{IDM}(v)$ ▷ Apply IDM following
    end if
  end for
  TTC Detection:
  Calculate TTC_{ramp} and TTC_{main} using (33).
  if $(\text{TTC}_{\text{ramp}} > \text{TTC}_{\text{safe}}) \wedge (\text{TTC}_{\text{main}} > \text{TTC}_{\text{safe}})$ **then**
    $a^* \leftarrow a_{\text{lane-change}}$
  else
    $a^* \leftarrow a_{\text{lane-keep}}$
  end if
end while

where $p_{\text{AV}}(t)$ and $p_{\text{NV}}(t)$ are the positions of the AV and an NV at time step t, respectively.

If there are overlaps between the AV's future trajectories and the NVs', the AV will use the IDM (29) to follow the nearest preceding vehicles and to synchronize with the overall vehicle flow. A penalty is imposed if the AV's velocity remains below the expected level over three seconds, encouraging timely progression toward the outlet while avoiding conflicts.

A safety margin (SM) is used to guide the selection of driving actions. As vehicles maintain a larger relative distance from each other while in proximity, the likelihood of their paths intersecting decreases. Therefore, the SM for each vehicle's maneuver is defined as

$$\text{SM} = \min_{a \in A_{\text{feasible}}} D_{\text{d},a,k} \tag{32}$$

where A_{feasible} is the set of feasible actions and $D_{\text{d},a,k}$ is the vehicular distance for action a at prediction time k.

At each decision point, the AV evaluates SMs for all actions and, if equal, prioritizes earlier lane changes to ease the burden on following AVs. The feasibility of lane-changing is evaluated using TTC defined by

$$\begin{aligned} \text{TTC}_{\text{ramp}} &= \frac{\text{Distance to HDV}_{\text{ramp}}}{\text{Speed of AV} - \text{Speed of HDV}_{\text{ramp}}} \\ \text{TTC}_{\text{main}} &= \frac{\text{Distance to HDV}_{\text{main}}}{\text{Speed of AV} - \text{Speed of HDV}_{\text{main}}} \end{aligned} \tag{33}$$

where TTC_{ramp} represents the TTC between the AV and the front HDV in the ramp lane, while TTC_{main} represents the TTC between the AV and the front HDV in the target main lane. *Distance to HDV$_{ramp}$* denotes the longitudinal distance between the AV and the front HDV in the ramp lane, and *Distance to HDV$_{main}$* represents the longitudinal distance between the AV and the front HDV in the target main lane. *Speed of HDV$_{ramp}$* refers to the longitudinal speed of the front HDV in the ramp lane, while *Speed of HDV$_{main}$* refers to the longitudinal speed of the front HDV in the target main lane. Both TTC_{ramp} and TTC_{main} are compared with the safe TTC. If both values are greater than the safe TTC, the lane change will be executed; otherwise, the lane-keeping command will be selected.

After each AV action, the next-highest priority action is considered in a loop until the AV safely reaches the main lane. This process helps avoid conflicts with NVs by replacing unsafe actions, as shown in Algorithm 2.

V. MPC for Enhancing DRL Performance

This section introduces MPC as a low-level controller for executing KL-MA2C decisions in real time, smoothing aggressive DRL commands into feasible actions. The hierarchical integration offers two main benefits: enforcing constraints with predictive optimization and serving as a safety filter under uncertainties. We use a prediction horizon of $N = 20$ steps (2 s) to balance computational efficiency with prediction accuracy while maintaining real-time performance requirements.

The MPC formulation is based on the uncertain vehicle dynamic model:

$$p_{\text{AV}}(t+1) = p_{\text{AV}}(t) + v_{\text{AV}}(t) \cdot \cos(h_{\text{AV}}(t)) \cdot \Delta t + w_p(t)$$
$$v_{\text{AV}}(t+1) = v_{\text{AV}}(t) + u(t) \cdot \Delta t + w_v(t)$$
$$h_{\text{AV}}(t+1) = h_{\text{AV}}(t) + \frac{v_{\text{AV}}(t)}{L} \cdot \tan(\delta(t)) \cdot \Delta t + w_h(t) \quad (34)$$

where $\Delta t = 0.1$ s is the sampling interval, $v_{\text{AV}}(t)$ is the speed, $h_{\text{AV}}(t)$ is the heading angle, $L = 2.8$ m is the wheelbase length, $u(t)$ is the acceleration, $\delta(t)$ is the steering angle, and $w_p(t)$, $w_v(t)$, $w_h(t)$ represent bounded model uncertainties.

To handle model uncertainties, the MPC enforces robust safety and feasibility constraints:

$$0 < v_{\text{AV}}(t) \le v_{\max},\ |u(t)| \le 2.0\,\text{m/s}^2,$$
$$|\delta(t)| \le 0.6\ \text{rad},\ \|p_{\text{AV}}(t) - p_{\text{NV}}(t)\|_2 \ge D_{\text{safe}} + \epsilon, \quad (35)$$

where $D_{\text{safe}} = 5$ m and $\epsilon = 1$ m provides a safety buffer against uncertainties.

At each time step t, the MPC control commands $u^*(t)$ and $\delta^*(t)$ are solved from the optimization problem:

$$\min \quad J_c$$
$$\text{s.t. (34), (35)},\ NV \in N_V,\ k \in [0, N-1] \quad (36)$$

with the cost function $J_c = \sum_{k=0}^{N-1}(v_{\text{AV}}(k) - v^*_{\text{AV}})^2 + \sum_{k=0}^{N-1}\sum_{i=1}^{N_v}(\|p^i_{\text{NV}}(k) - p^i_{\text{AV}}(k)\|_2 - D_{\text{safe}})^2 + \lambda \sum_{k=0}^{N-1} u^2(k)$, where $v^*_{\text{AV}}(k)$ is the target speed, N_V is the number of NVs, and λ is a weighting factor. The MPC optimization problem is solved using the Computer Algebra System with Automatic Differentiation for Optimization (CasADi) framework with the IPOPT solver [37]. CasADi provides efficient automatic differentiation and sparse optimization capabilities, enabling real-time MPC solutions. IPOPT is tuned for this application with a 100-iteration limit and 1.0e−6 tolerance to balance accuracy and efficiency.

The complete control algorithm (see Algorithm 3) employs a hierarchical structure, with MPC serving as the primary controller and PID as a fallback. The PID is designed as

$$u_{\text{PID}}(t) = K_p(v^*_{\text{AV}} - v_{\text{AV}}(t)) + K_i \int_0^t (v^*_{\text{AV}} - v_{\text{AV}}(\tau))\mathrm{d}\tau + K_d \frac{\mathrm{d}}{\mathrm{d}t}(v^*_{\text{AV}} - v_{\text{AV}}(t)) \quad (37)$$

where K_p, K_i and K_d are tuned to provide stable control while maintaining safety requirements.

If an optimal solution to the MPC problem is found within the specified time limit (30 ms), it is applied to the AV; otherwise, the system smoothly transitions to a PID controller that tracks the target velocity. This hierarchical architecture ensures continuous operation even under computational constraints or numerical difficulties in the MPC optimization.

Algorithm 3: MPC With PID Fallback.

Input: v^*_{AV}: Target speed from KL-MA2C; D_{safe}: Safe distance threshold; N: Prediction horizon
Output: $u^*(t)$: Optimal acceleration control input
$\delta^*(t)$: Optimal steering control input

```
MPC_Controller(v*_AV)
Initialize CasADi optimizer
NV ← get_surrounding_vehicles()
Define decision variables u(k), δ(k) for k ∈ [0, N − 1]
J_c ← 0
for k ← 0 to N − 1 do
    Predict NV states and update AVs dynamics using (34)
    J_c ← J_c + (v_AV(k) − v*_AV)^2
    for vehicle in NV do
        J_c ← J_c + (‖p_NV(k) − p_AV(k)‖_2 − D_safe)^2
    end for
    J_c ← J_c + λu^2(k)
    Add constraints from (35)
end for
Solve optimization with objective J_c under 30 ms time limit
if optimization successful then
    return u*(t), δ*(t)
else
    u*(t) ← PID_Controller(v*_AV, v_AV) using (37)
    δ*(t) ← Safety_Steering()
    return u*(t), δ*(t)
end if
```

VI. Computational Complexity Analysis

The KL-MA2C framework introduces computational complexity at multiple levels. For the neural network component, with L layers each having N neurons, the KAN-enhanced architecture has a computational complexity of $\mathcal{O}(LN^2 + LNS)$ for forward propagation, where S is the number of spline segments in the activation functions. The LSTM layers, processing sequences of length T, add $\mathcal{O}(4NT)$ operations for the four gates and cell state updates. The action inspector performs safety checks with complexity $\mathcal{O}(K)$, where K is the number of surrounding vehicles, as it computes TTC and safety distances for each nearby vehicle. This lightweight computation takes approximately 2–3 ms. The MPC optimization is the most computationally intensive component. For a prediction horizon of N steps and P decision variables, the IPOPT solver has complexity $\mathcal{O}(NP^2)$. Warm-starting from previous solutions reduces the average number of iterations needed for convergence. The sparse matrix operations in CasADi further improve computational efficiency.

Our empirical measurements demonstrate that KL-MA2C network inference requires approximately 5 ms, the action inspector adds 3 ms of processing time, and the MPC optimization completes within 25–30 ms, resulting in a total processing time under 40 ms . This performance enables reliable 10Hz control updates, meeting the real-time requirements for autonomous driving. The hierarchical control structure further ensures safety is maintained even when computational constraints necessitate temporary use of the PID controller.

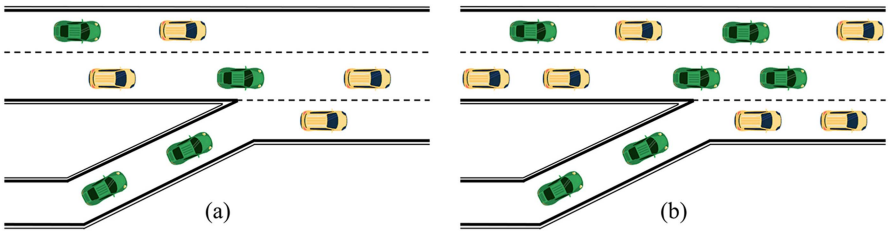


Fig. 5. Ramp scenarios for validation. (a) Normal mode and (b) Hard mode.

TABLE I
PARAMETERS FOR THE KL-MA2C AGENT

Parameter	Default Value	Description
n_episodes	10000	Total training episodes
warm_up_episodes	200	Warm-up episodes
gamma	0.99	Discount
train_steps	5	Train steps
target_update_freq	100	Target update
learning_rate	0.001	Learning rate
replay_buffer_cap	50000	Replay capacity
batch_size	100	Batch size
width	64	Hidden width
grid	10	Grid size
eval_freq	200	Evaluation frequencies
eval_episodes	5	Evaluation episodes
seeds	[20, 40, 60]	Random seeds
sim_frequency	15	Simulation frequencies
duration	20	Duration
policy_frequency	5	Policy frequencies
collision_reward	-20	Collision reward
is_arrived_reward	15	Arrive reward
high_speed_reward	4	High speed reward
change_lane_cost	0.2	Penalty change lanes
headway_cost	4	Headway cost
headway_time	1.2	Headway time
action_inspector	True	Activate action inspector
low_level_control	MPC	Choose MPC or PID
eval_seeds	[10, 30, 50]	Evaluation seeds

VII. SIMULATION RESULTS

The proposed KL-MA2C is evaluated on reward, collision rate, and speed of AVs on three-lane ramps. Its generalizability is tested under varying traffic conditions, as shown in Fig. 5, with HDV initial positions and speeds randomly set within a small range. Given the risks and regulatory constraints of real-world testing, scenario-based virtual validation is employed for its safety, reproducibility, and efficiency. All simulations are performed on the Highway platform [38], using Python 3.6, PyTorch 1.10.0, Ubuntu 20.04, an Intel i5-12600KF CPU, RTX 3090 GPU, and 64 GB RAM. All experiments are conducted using three random seeds, with mean results plotted and standard deviations shown as shaded areas.

The environment settings and parameters are listed in Table I. The key hyperparameters were selected based on standard reinforcement learning practices and preliminary tuning experiments [23]. The learning rate of 0.001 provides stable convergence, while the discount factor of 0.99 ensures appropriate future reward consideration. The KAN grid size of 10 and hidden width of 64 balance network expressiveness with computational efficiency. Safety-related parameters were set conservatively to prioritize collision avoidance. These parameter choices are robust, as shown by consistent performance across random seeds and traffic scenarios, with small standard deviations in the shaded regions of our results.

Two types of comparisons are conducted for validation. First, KL-MA2C is compared with four MA2C variants incorporating LSTM, MPC, and action inspection, respectively. Second, it is benchmarked against four state-of-the-art (SOTA) DRL algorithms—MADQN, MADDPG, MAACKTR, and MAPPO—under two traffic flow settings. The evaluation scenarios are summarized as follows:

- *Ablation study in normal mode (Section VII-A):* The proposed KL-MA2C (*Ours*) is compared with MA2C+KAN+LSTM+Action Inspector (*Ours: No MPC*), MA2C+KAN+Action Inspector+MPC (*Ours: No LSTM*), and MA2C+KAN+LSTM+MPC (*Ours: No Inspector*).
- *Normal mode validation (Section VII-B):* The proposed KL-MA2C is compared with benchmark algorithms with four HDVs and four AVs on the ramp as shown in Fig. 5(a).
- *Hard mode validation (Section VII-C):* The proposed KL-MA2C is compared with benchmark algorithms with six HDVs and six AVs on the ramp as shown in Fig. 5(b).

A. Ablation Study

This section reports the experiments for evaluating the crucial functions of LSTM, action inspector, and MPC of the proposed system in normal mode. Fig. 6 presents the results of reward, mean driving speed, and collision rates.

As shown in Fig. 6(a), the proposed KL-MA2C (*Ours*) demonstrates superior learning performance with stable convergence to approximately 225 reward points after only 1500 episodes. In contrast, the ablation variants show distinct learning characteristics: *Ours: No MPC* achieves similar reward levels but with delayed convergence around 5000 episodes, *Ours: No LSTM* exhibits fluctuations throughout training with high variance, and *Ours: No Inspector* shows the poorest performance with lower reward ceiling around 170 and persistent instability. This superior convergence demonstrates the synergistic effect of KAN's adaptive function approximation with LSTM's temporal modeling capabilities, while the stability indicates that all components contribute to robust learning.

As shown in Fig. 6(b), the proposed KL-MA2C (*Ours*) maintains consistent high-speed performance at 23 $\mathrm{m/s}$ with minimal variance throughout training. Notably, *Ours: No LSTM* shows

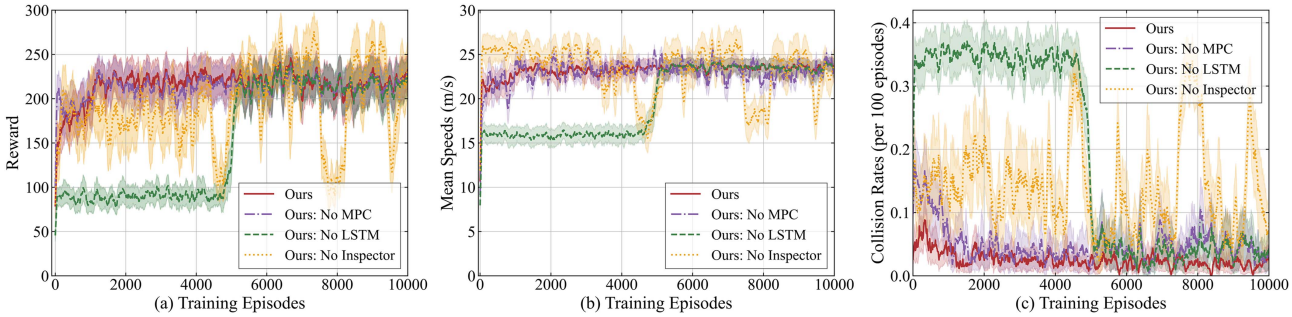


Fig. 6. Functional performance comparison during training in normal mode. (a) Reward, (b) Speed, (c) Collision rate.

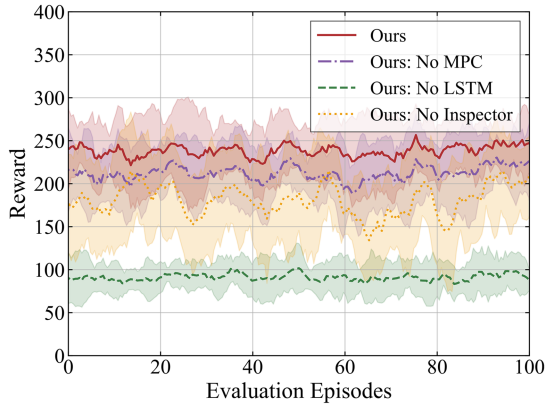


Fig. 7. Evaluation rewards for ablation study in normal mode.

initially lower speeds around 16 $\mathrm{m/s}$ before gradually improving, suggesting that LSTM's temporal modeling is crucial for efficient speed control. Both *Ours: No MPC* and *Ours: No Inspector* achieve similar final speeds but with higher variance, indicating less stable control. The speed stability (23 $\mathrm{m/s}$ with variance of 2) indicates that *Ours* maintains consistent performance while other variants show either delayed learning or higher variance patterns.

As shown in Fig. 6(c), the collision rate analysis provides the most compelling evidence for our framework's safety benefits. The proposed KL-MA2C (*Ours*) achieves collision rates below 0.05 per 100 episodes after convergence, with rapid improvement visible within the first 2000 episodes. The ablation study reveals critical insights: *Ours: No Inspector* shows dramatically higher collision rates (0.15-0.25 per 100 episodes), confirming the action inspector's essential role in safety assurance. *Ours: No LSTM* and *Ours: No MPC* both exhibit intermediate collision rates around 0.15 and 0.08 per 100 episodes respectively, with higher variance indicating less reliable safety performance. The sharp drop in collision rate (0.2 to 0.05) confirms that combining the action inspector and MPC adds crucial robustness, while rapid convergence indicates effective safety learning.

Fig. 7 presents the rewards of the proposed KL-MA2C and three other variants using the converged policies. It is seen that the proposed KL-MA2C (*Ours*) has a higher reward than the variants after convergence. More specifically, the rewards of *Ours*, *Ours: No MPC*, *Ours: No LSTM*, and *Ours: No Inspector* are around 250, 200, 175, and 100, respectively. This demonstrates the effectiveness of the proposed KL-MA2C in terms of safety and efficiency.

TABLE II
COMPARISON OF COLLISION RATE AND AVERAGE SPEED IN NORMAL MODE

Algorithm	**Collision rate** (%)	**Mean speed** ($\mathrm{m/s}$)
MAPPO	6.1 ± 0.05	18.56 ± 0.52
MADDPG	6.5 ± 0.06	22.57 ± 0.55
MAACKTR	6.6 ± 0.8	17.39 ± 0.41
MADQN	5.7 ± 0.07	18.31 ± 0.43
Ours	$\mathbf{1.76 \pm 0.03}$	$\mathbf{23.39 \pm 0.25}$

B. Comparison With Benchmarks in Normal Mode

We evaluate the safety and efficiency of our system in normal mode against SOTA DRL benchmarks: MADDPG, MAPPO, MAACKTR, and MADQN.

Fig. 8 shows training performance—reward, speed, and collision rate—of the five methods. As shown in Fig. 8(a), during training the proposed KL-MA2C reaches a higher reward compared to the four baselines. The reward gap (225 vs 175–180 for benchmarks) demonstrates KL-MA2C's superior learning efficiency in multi-agent environments. As shown in Fig. 8(b), our KL-MA2C achieves a higher average speed compared to all baselines. The average speed and variance of the KL-MA2C, MAPPO, MADDPG, MAACKTR, and MADQN are $(23, 1)$, $(18, 2)$, $(21, 3)$, $(18, 2)$, and $(19, 2)$, respectively. These consistent performance gaps indicate that KAN's enhanced function approximation and LSTM's temporal modeling translate effectively to practical driving scenarios. As shown in Fig. 8(c), our KL-MA2C has the lowest collision rate. The collision rates are below 0.055 per 100 episodes for KL-MA2C compared to 0.1–0.125 for SOTA benchmarks. This safety advantage reflects the effectiveness of our integrated approach in handling human-vehicle interactions.

Fig. 9 presents their evaluation rewards using the converged policies. It is seen that the proposed KL-MA2C achieves a higher evaluation reward compared to the benchmarks, demonstrating the effectiveness of KL-MA2C in enhancing driving safety and efficiency.

Table II compares the average speed and collision rate of the proposed KL-MA2C with four SOTA benchmarks. The proposed KL-MA2C achieves the lowest collision rate at $1.76 \pm$

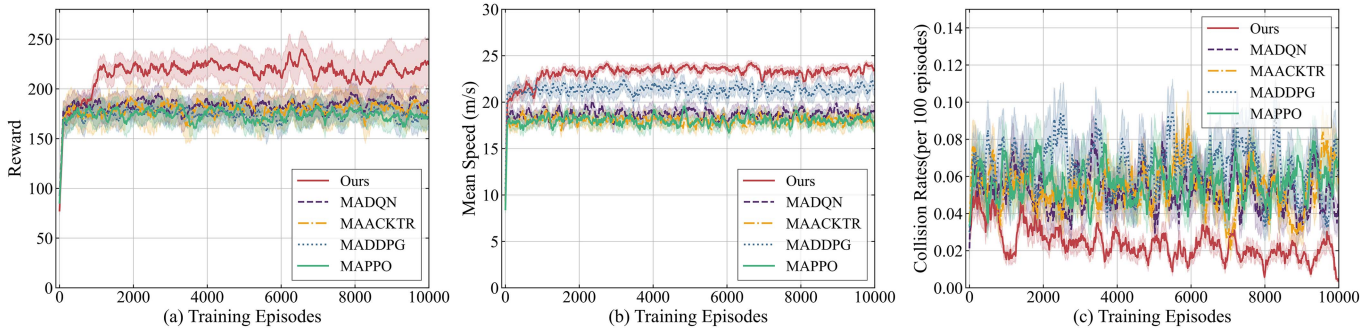


Fig. 8. Performance comparison during training in normal mode. (a) Reward, (b) Speed, and (c) Collision rate.

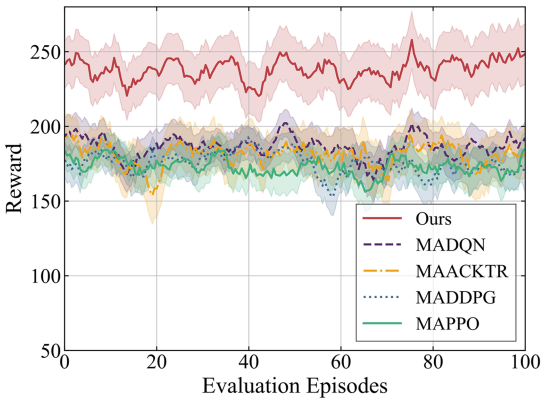


Fig. 9. Comparison of evaluation rewards in normal mode.

0.03 %, followed by MADQN at 5.7 ± 0.07 %, MAPPO at 6.1 ± 0.05 %, MADDPG at 6.5 ± 0.06 %, and MAACKTR at 6.6 ± 0.8 %, indicating the safety performance of the KL-MA2C. In addition, the KL-MA2C has the highest average speed of 23.39 ± 0.25 m/s, which is larger than that of MADDPG at 22.57 ± 0.55 m/s, MAPPO at 18.56 ± 0.52 m/s, MADQN at 18.31 ± 0.43 m/s, and MAACKTR at 17.39 ± 0.41 m/s, showcasing greater efficiency.

C. Comparison With Benchmarks in Hard Mode

This section evaluates effectiveness of the proposed KL-MA2C (*Ours*) in hard mode—a more challenging scenario involving six HDVs and six AVs on the ramp (see Fig. 5(b)).

Fig. 10 presents the reward, driving speed, and collision rates of the proposed KL-MA2C (*Ours*), and the baselines (MAPPO, MADDPG, MAACKTR, and MADQN) during training under hard mode settings. As shown in Fig. 10(a), the proposed KL-MA2C reaches a higher reward compared to the four baselines. The rewards of the proposed KL-MA2C, MAPPO, MADDPG, MAACKTR, and MADQN are around 200, 85, 125, 175, and 125, respectively. The dramatic reward differences (200 vs 85-175) highlight KL-MA2C's superior scalability in complex scenarios where traditional DRL algorithms struggle with increased interaction complexity. As shown in Fig. 10(b), the proposed KL-MA2C achieves a higher average speed with better

TABLE III
COMPARISON OF COLLISION RATE AND AVERAGE SPEED IN HARD MODE

Algorithm	**Collision rate** (%)	**Mean speed** (m/s)
MAPPO	22.57 ± 0.04	20.56 ± 1.15
MADDPG	7.73 ± 0.06	20.36 ± 0.85
MAACKTR	10.47 ± 0.5	16.83 ± 0.41
MADQN	8.62 ± 0.04	16.92 ± 0.43
Ours	$\mathbf{1.38 \pm 0.02}$	$\mathbf{22.82 \pm 0.21}$

stability. The maintained high speed (22.5 m/s) under challenging conditions demonstrates the framework's robustness through integrated safety mechanisms. As shown in Fig. 10(c), the proposed KL-MA2C has significantly lower collision rates (0.055 per 100 episodes vs 0.1–0.5 for benchmarks). Notably, the collision rate in hard mode (1.38 %) is slightly lower than that in normal mode (1.76 %), which reflects the philosophical insight that increased AV density enhances collective intelligence and cooperative behavior. This counter-intuitive result demonstrates that multi-agent systems can achieve emergent safety properties through coordinated decision-making that individual agents cannot accomplish alone. This performance gap becomes more pronounced in high-density scenarios, confirming that KAN-LSTM integration provides particular benefits when traditional neural architectures reach their approximation limits.

Fig. 11 presents the evaluation rewards of the proposed KL-MA2C and four baseline algorithms—MAPPO, MADDPG, MAACKTR, and MADQN—under the hard mode setting, using their respective converged policies. It can be seen that the proposed KL-MA2C achieves the highest reward during evaluation, compared to all the baselines. A comparison of their average speeds and collision rates is presented in Table III. The collision rate of the KL-MA2C is 1.38 ± 0.02 %, smaller than that of MAPPO at 22.57 ± 0.04 %, MADDPG at 7.73 ± 0.06 %, MAACKTR at 10.47 ± 0.5 %, and MADQN at 8.62 ± 0.04 %. Additionally, the average speed of the KL-MA2C is 22.82 ± 0.21 m/s, greater than MAPPO at 20.56 ± 1.15 m/s, MADDPG at 20.36 ± 0.85 m/s, MAACKTR at 16.83 ± 0.41 m/s, and MADQN at 16.92 ± 0.43 m/s. These quantitative results reveal that KL-MA2C's advantages become more pronounced under challenging conditions, with up to 84 % collision reduction while maintaining superior speeds, indicating the particular effectiveness of our architectural innovations in safety-critical multi-agent scenarios.

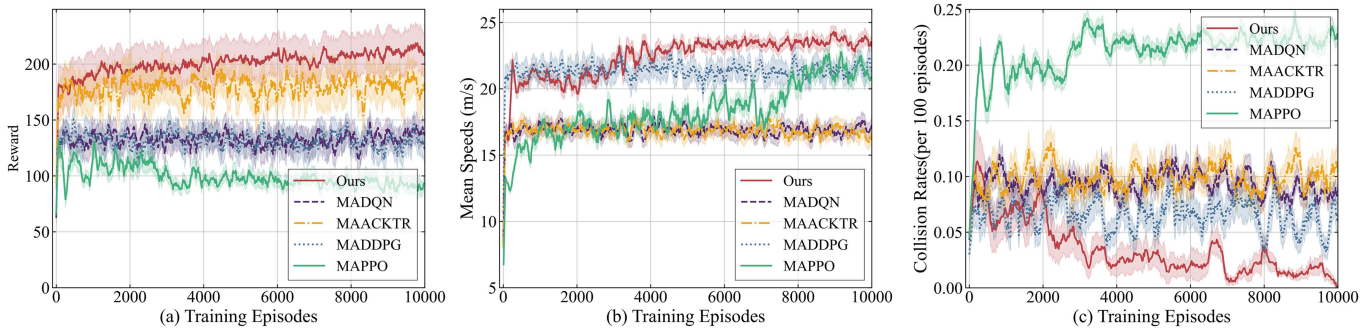


Fig. 10. Performance comparison during training in hard mode. (a) Reward, (b) Speed, and (c) Collision rate.

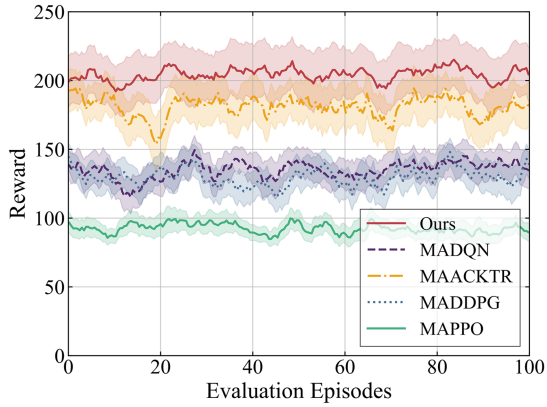


Fig. 11. Comparison of evaluation rewards in hard mode.

Overall, all the simulation results demonstrate the proposed KL-MA2C's advantages in balancing safety and efficiency under varying vehicle flows, and show its practical applicability in complex driving scenarios.

VIII. Conclusion and Future Work

This paper presents a novel KAN-LSTM enhanced Multi-Agent Advantage Actor-Critic (KL-MA2C) algorithm for autonomous ramp merging, making three key contributions: 1) Synergistic integration of KAN with LSTM for superior function approximation and temporal behavior modeling; 2) An action inspector mechanism enforcing safety constraints through TTC analysis; and 3) Hierarchical control architecture combining MPC optimization with computational robustness. Extensive experiments demonstrate substantial performance improvements over baselines, achieving up to 84 % collision reduction (from 22.57 % to 1.38 % in high-density scenarios) while maintaining 11–35 % higher average speeds and 80 % faster convergence compared to SOTA DRL benchmark algorithms (MAPPO, MADDPG, MAACKTR, MADQN). These results establish KL-MA2C as an effective solution for autonomous ramp merging, with clear implications for safe and efficient autonomous driving in complex multi-agent environments. Future work could explore extending the framework to more complex traffic scenarios such as urban intersections and multi-lane highways, investigating integration with other autonomous driving subsystems, and conducting real-world validation trials to assess practical deployment feasibility.

References

[1] T. M. Rajeh, Z. Luo, M. H. Javed, F. Alhaek, and T. Li, "A clustering-based multi-agent reinforcement learning framework for finer-grained taxi dispatching," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11269–11281, Sep. 2024.

[2] X. Chen, Y. Sun, T. Zhang, X. Wang, S. Xiong, and K. Cao, "An anytime trajectory optimizer for accurately parking an autonomous vehicle in tiny spaces," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 8772–8783, Jun. 2025.

[3] G. Zhang, J. Xie, B. Peng, H. Zhang, and D. Song, "Power allocation strategy of multi-static radar network tracking maneuvering jammer based on Stackelberg game," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 9767–9776, Jun. 2025.

[4] W. Yi, Z. Xu, Y. Wu, Y. Jiang, and Z. Yao, "Fundamental diagram modeling of mixed traffic flow considering dedicated lanes for human-driven vehicles," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 8808–8823, Jun. 2025.

[5] Z. Lin et al., "A conflicts-free, speed-lossless KAN-based reinforcement learning decision system for interactive driving in roundabouts," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 25, 2025, doi: 10.1109/TITS.2025.3578279.

[6] Y. Bie, Y. Ji, and D. Ma, "Multi-agent deep reinforcement learning collaborative traffic signal control method considering intersection heterogeneity," *Transp. Res. Part C: Emerg. Technol.*, vol. 164, 2024, Art. no. 104663.

[7] K. Cai, Z. Li, T. Guo, and W. Du, "Multiairport departure scheduling via multiagent reinforcement learning," *IEEE Intell. Transp. Syst. Mag.*, vol. 16, no. 2, pp. 102–116, Mar./Apr. 2024.

[8] 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/Jun. 2024.

[9] M. Pourabdollah, E. Bjärkvik, F. Fürer, B. Lindenberg, and K. Burgdorf, "Calibration and evaluation of car following models using real-world driving data," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–6.

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

[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.

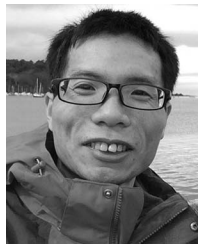
[12] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[13] 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.

[14] F. Altché and A. de L. Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 353–359.

[15] Y. Jeong, "Predictive lane change decision making using bidirectional long shot-term memory for autonomous driving on highways," *IEEE Access*, vol. 9, pp. 144985–144998, 2021.
[16] A. Khan, M. M. Fouda, D.-T. Do, A. Almaleh, and A. U. Rahman, "Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 11, pp. 94371–94391, 2023.
[17] T. Chen, Y. Zhang, X. Qian, and J. Li, "A knowledge graph-based method for epidemic contact tracing in public transportation," *Transp. Res. Part C, Emerg. Technol.*, vol. 137, 2022, Art. no. 103587.
[18] Y. Gao, D. Li, Z. Sui, and Y. Tian, "Trajectory planning and tracking control of autonomous vehicles based on improved artificial potential field," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12468–12483, Sep. 2024.
[19] P. Hang, C. Huang, Z. Hu, Y. Xing, and C. Lv, "Decision making of connected automated vehicles at an unsignalized roundabout considering personalized driving behaviours," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4051–4064, May 2021.
[20] Z. Han, P. Chen, B. Zhou, and G. Yu, "Real-time navigation of unmanned ground vehicles in complex terrains with enhanced perception and memory-guided strategies," *IEEE Trans. Veh. Technol.*, vol. 74, no. 3, pp. 3723–3735, Mar. 2025.
[21] 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.
[22] Z. Peng, Y. Wang, L. Zheng, and J. Ma, "Bilevel multi-armed bandit-based hierarchical reinforcement learning for interaction-aware self-driving at unsignalized intersections," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 8824–8838, Jun. 2025.
[23] K. Yang, X. Tang, S. Qiu, S. Jin, Z. Wei, and H. Wang, "Towards robust decision-making for autonomous driving on highway," *IEEE Trans. Veh. Tech.*, vol. 72, no. 9, pp. 11251–11263, Sep. 2023.
[24] 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.
[25] P. Wang, C.-Y. Chan, and H. Li, "Automated driving maneuvers under interactive environment based on deep reinforcement learning," 2018, *arXiv:1803.09200*.
[26] 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. 11285–11297, Jun. 2025.
[27] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.
[28] 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.
[29] Z. Tian et al., "Efficient and balanced exploration-driven decision making for autonomous racing using local information," *IEEE Trans. Intell. Veh.*, vol. 10, no. 3, pp. 1732–1748, Jul. 23, 2024, doi: 10.1109/TIV.2024.3432713.
[30] K. Moghaddasi et al., "An advanced deep reinforcement learning algorithm for three-layer D2D-edge-cloud computing architecture for efficient task offloading in the Internet of Things," *Sustain. Comput.: Inform. Syst.*, vol. 43, 2024, Art. no. 100992.
[31] K. Moghaddasi et al., "An enhanced asynchronous advantage actor-critic-based algorithm for performance optimization in mobile edge computing-enabled Internet of Vehicles networks," *Peer-to-Peer Netw. Appl.*, vol. 17, no. 3, pp. 1169–1189, 2024.
[32] K. Moghaddasi, S. Rajabi, F. S. Gharehchopogh, and M. Hosseinzadeh, "An energy-efficient data offloading strategy for 5G-enabled vehicular edge computing networks using double deep Q-network," *Wireless Pers. Commun.*, vol. 133, no. 3, pp. 2019–2064, 2023.
[33] A. M. Rahmani et al., "An optimizing GEO-distributed edge layering with double deep q-networks for predictive mobility-aware offloading in mobile edge computing," *Ad Hoc Netw.*, vol. 172, 2025, Art. no. 103804.
[34] K. Moghaddasi, S. Rajabi, and F. S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled IoV-MEC systems: A double deep Q-network approach," *IEEE Access*, vol. 12, pp. 3437–3463, 2024.
[35] 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.
[36] Z. Liu et al., "KAN: Kolmogorov-Arnold networks," 2024, *arXiv:2404.19756*.
[37] X. Qi, L. Zhang, P. Wang, J. Yang, T. Zou, and W. Liu, "Learning-based MPC for autonomous motion planning at freeway off-ramp diverging," *IEEE Trans. Intell. Veh.*, vol. 9, no. 10, pp. 6183–6194, Oct. 2024.
[38] E. Leurent, "An environment for autonomous driving decision-making," *GitHub Repository*, Accessed: Dec. 12, 2024. [Online]. Available: https://github.com/eleurent/highway-env

Zhihao Lin received the M.S. degree from the College of Electronic Science & Engineering, Jilin University, Jilin, China. He is currently working toward 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.



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

Xianxian Zhao (Member, IEEE)received the Ph.D. degree in electrical engineering from the University of Birmingham, Birmingham, U.K., in 2018. She is currently the Principle Investigator of the SEAI-RD&D Programme with University College Dublin, Dublin, Ireland. Her research interests include modelling and control of electric machines and converters, stability and power quality of power-electronic-based power systems, and model order reduction of large power systems.