Learning-based Lane Selection and Driving Orders for Connected Automated Vehicles at Multi-lane Freeway Merging Sections

Jieming Chen, Yifeng Zhang, Yue Zhou, Yiwei Wu, Edward Chung, Guillaume Sartoretti

***Abstract*—Cooperative control of connected automated vehicles (CAVs) offers a promising solution for reducing traffic congestion and accidents. However, existing optimization-based and search-based methods for trajectory planning and vehicle scheduling struggle with real-time multi-vehicle control. This paper introduces a hybrid bi-level approach that nests optimization modelling within deep reinforcement learning (DRL) to jointly optimize vehicle sequences, lane selections, and trajectories, providing a rapid, safe, and high-quality solution to enhance traffic performance at multi-lane freeway merging sections. Specifically, we approach the problem of lane selection and vehicle sequencing for multiple vehicles as a multi-step decision-making process. At the upper level, we design a DRL agent with an attention-based encoder-decoder structure that auto-regressively constructs driving sequences and lane choices. It generates a probability matrix to select the next passing vehicle and target lane based on prior decisions at each step. The attention mechanism enables the centralized upper level to adapt to scenarios with varying vehicle counts without the need to retrain. At the lower level, we formulate a model predictive control (MPC) planner to generate safe trajectories. The resulting travel delay guides the upper-level DRL agent learning to maximize overall traffic efficiency. Moreover, we introduce a leader-and-lane-specific credit assignment mechanism that leverages domain knowledge to link each action with associated travel delays. This mechanism enables the agent to accurately recognize the impact of decisions on total delay, enhancing learning performance. Simulation results suggest that the proposed approach's superior real-time performance and scalability from several to over a dozen vehicles, making it well-suited for practical automated merging tasks.**

***Index Terms*—Connected automated vehicles, deep reinforcement learning, multi-lane on-ramp merging.**

I. Introduction

FREEWAY on-ramp merging areas are commonly recognized as typical bottlenecks where multiple traffic streams compete for limited roadway capacity. The complex vehicle interaction in these lane-drop areas often results in excessive emissions, congestion, and accidents [1]–[3]. Widely implemented control approaches focus on the macroscopic management of traffic flow, e.g. variable speed limit [4] and ramp metering [5]. However, these approaches cannot directly coordinate individual vehicle behaviour. In the future, connected automated vehicles (CAVs) may introduce many developments in microscopic traffic control, such as platoon formation [6] and cooperative lane changes [7]. Leveraging the cooperative driving capabilities of CAVs is expected to enhance safety and improve traffic performance [8]–[11].

Significant research effort has been devoted to CAV-based merging control. Early studies have mainly focused on trajectory planning with predefined vehicle sequences for single-lane freeway merging. These studies highlight the importance of coordinated speed adjustments between mainline and on-ramp vehicles for smooth merging maneuvers [12]–[17]. Subsequent research has shifted towards vehicle sequence decision-making, i.e., determining the order in which vehicles pass through conflict areas [18]–[23]. In more general multi-lane freeway merging scenarios, the joint optimization of lane selection, vehicle sequences, and trajectories becomes more challenging [24]–[31]. Existing approaches typically fall into three categories: search-based, optimization-based, and learning-based methods. However, search-based methods often struggle to ensure solution quality, especially as the number of CAVs increases and the solution space expands exponentially. Optimization-based approaches are often too computationally demanding for real-time use. Learning-based methods typically lack formal safety guarantee.

To simultaneously ensure safety, maintain high solution quality in joint optimization, and meet strict real-time requirements, we propose a hybrid bi-level control framework, where a deep reinforcement learning (DRL) agent at the upper level determines lane selection and vehicle sequences and a model predictive control (MPC) planner at the lower level generates trajectories to implement upper-level's decisions. This proposed hybrid approach leverages the complementary strengths of DRL's computational efficiency and optimization techniques' safety guarantee to control multiple vehicles driving through merging sections. Specifically, we frame lane selection and vehicle sequencing as a multi-step decision-making process, where the DRL agent generates a probability matrix for target vehicles and lanes at each step based on previous vehicle and lane decisions. To achieve this, we design a policy network with an encoder-decoder structure as a DRL

This research was supported by the General Research Fund #15207320 (InCoMe) of the University Grants Committee of Hong Kong and A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002). (*Corresponding author: Edward Chung*)

Jieming Chen, Yue Zhou, and Edward Chung are with Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: jieming.chen@connect.polyu.hk, zhouyue30@msn.com, edward.chung@polyu.edu.hk).

Yifeng Zhang and Guillaume Sartoretti are with Department of Mechanical Engineering, College of Design and Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore (e-mail: yifeng@u.nus.edu, guillaume.sartoretti@nus.edu.sg).

Yiwei Wu is with Department of Logistics and Maritime, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: yi-wei.wu@polyu.edu.hk).

The code is available at https://github.com/DRL_CAV_Merge.git.

agent that auto-regressively constructs vehicle sequences and lane selections. We then formulate an MPC planner to generate trajectories and use the corresponding travel time as reward signals to guide the DRL agent in learning a strategy that minimizes total travel time. That is, the lower level performs planning based on decisions from the upper level, and the upper level optimizes its decisions based on the results of the lower level. Moreover, we propose a leader-and-lane-specific credit assignment mechanism that associates the decision made at each step with the travel delay of affected vehicles. This credit assignment mechanism helps the agent recognize the impact of decisions on total delay, thereby improving learning performance. We compare our approach with various methods, and empirical simulation results show that it outperforms rule-based methods and achieves performance comparable to metaheuristic methods that require tens of minutes for search, showing superior real-time performance of our approach. Moreover, we evaluate our approach under varying numbers of vehicles, from several to over a dozen, showing its great scalability. We believe that these attributes, solid real-time performance and scalability to varying vehicle counts, make our approach highly suited for practical application.

The contributions of this paper are summarized as follows.

1) We propose a hybrid bi-level approach that integrates DRL and MPC to optimize lane selection, driving sequences, and trajectories, enabling real-time planning for multiple vehicles to maximize traffic efficiency.
2) We design a DRL agent that auto-regressively constructs vehicle sequences and lane selection. At each step, the DRL agent generates a vehicle-lane probability matrix based on previous decisions to determine the next passing vehicle and its target lane.
3) We propose a leader-and-lane-specific credit assignment mechanism that links each action with the related travel delays, aiding the DRL agent in recognizing the impact of decisions and enhancing learning performance.
4) We demonstrate the superior real-time performance of the proposed hybrid approach, including traffic efficiency and computation time, compared to the baseline methods. The effectiveness of the proposed leader-and-lane-specific credit assignment mechanism and the adopted DRL techniques is also validated.

II. Literature Review

CAV-based merging control approaches can be categorized into three types: tree-search-based, optimization-based, and learning-based methods.

A. Tree-search-based Methods

Vehicle sequences are represented as paths from the root to the leaf nodes of a tree, where each node corresponds to a CAV, and child nodes are generated by adding uncovered CAVs at successive levels. Many search methods are employed to explore optimal paths within this tree structure for single-lane freeway merging problems. Pei et al. [18] applied dynamic programming to find the optimal path, though travel delay for each node was estimated rather than derived from actual trajectories. Tang et al. [19] utilized the monte carlo tree search to find a sub-optimal sequence within a limited time. Shi et al. [20] used the depth-first search with heuristic pruning rules to obtain solutions. Chen et al. [21] proposed a sequential search method that incrementally adds one on-ramp vehicle to the tree at a time, significantly reducing the number of branches. Xie et al. [22] designed a Tabu Search algorithm to determine merging sequences.

B. Optimization-based Mmethods

The problem is formulated as a mixed integer programming model, with integer variables representing discrete sequencing and lane-changing decisions and continuous variables denoting acceleration, velocity, and position. Chen et al. [24] formulated an optimization model incorporating car-following, cruising, and lane-changing driving modes. The possible driving modes of multiple vehicles were enumerated and fed into the model to find the best solution. Dollar et al. [25] proposed a mixed-integer quadratic program model to plan a vehicle's lane change and acceleration, integrating the analytic optimal control and integer programming. Yang et al. [26] modeled the merging problem as a cooperative game, where different sequences and lane assignments are enumerated and compared for overall traffic performance. In contrast, Yu et al. [27] and Wei et al. [28] approached the problem as a non-cooperative game, optimizing the benefit of an individual vehicle. Although these models explicitly define the problem to seek an optimal solution, the combinatorial explosion of possible sequences and lane assignments makes these models computationally intractable, posing significant challenges in meeting real-time requirements.

C. Learning-based Methods

Through offline training, DRL-based methods learn policies that are applied online to solve the problem efficiently. One DRL-based approach controls each CAV's low-level actions (e.g., acceleration or velocity) or high-level actions (e.g., yielding and lane changes). Chen et al. [29] formulated the mixed-traffic merging problem as a multi-agent reinforcement learning problem, where each vehicle makes high-level decisions, such as turning left, turning right, cruising, speeding up, and slowing down. Hu et al. [30] employed a graph convolutional network with attention to capture high-dimensional CAV features and designed an action space containing discrete manoeuvres, including acceleration, deceleration, and lane changes. Hwang et al. [31] proposed a finite state machine for four phases (gap selection, gap approach, negotiation, and lane-change execution) and used DRL to execute lane changes. Another approach treats the problem of CAV navigation through conflict areas as a combinatorial optimization problem, inspired by recent advancements in applying DRL to solve vehicle routing problems [32]–[34]. Zhang et al. [35] adopted the pointer network structure to determine a driving sequence crossing an intersection and used tree search to further refine the solutions. Similarly, Jiang et al. [36] applied this structure to the single-lane freeway merging scenario. This type of approach uses a sequence-to-sequence network

combined with DRL to optimize vehicle passing sequences, providing a safe alternative to speed-optimization methods.

D. Comparison of Existing and Proposed Methods

In comparison, tree-search-based methods are flexible and effective, but their solution quality often deteriorates as the problem size increases. Optimization-based methods obtain optimal solutions, but their high computational demands make real-time application challenging. Learning-based methods allow rapid inference after training, yet they generally lack formal safety guarantees and may not reliably prevent unsafe actions, especially in unexpected or unfamiliar scenarios.

Overall, existing work on joint decisions for lane selections, vehicle sequences, and vehicle trajectories remains limited and faces significant downsides in terms of solution quality, real-time requirement, and decision safety. Hence, this paper proposes a bi-level control framework that combines the complementary strengths of DRL for real-time performance and optimization techniques for safety guarantee. Moreover, our DRL-based agent formulates decision-making as lane assignment and vehicle sequencing, enabling a novel high-level coordination strategy. In contrast to conventional learning-based methods that directly output low-level control actions, our abstraction enables more structured planning and achieves more efficient, globally coordinated, and safer actions.

III. Mathematical Problem Formulation

As shown in Fig. 1, we consider a typical freeway on-ramp merging section, which includes two mainline lanes (an inside lane and an outside lane), an on-ramp lane extending into an acceleration lane, a roadside unit (RSU), and a trigger point. The RSU, positioned upstream of the merge gore, collects information from CAVs and transmits commands to them. The trigger point on the on-ramp lane activates the proposed control when on-ramp vehicles approach. Before any vehicle approaches the trigger point, all CAVs operate in car-following mode. Once an on-ramp vehicle crosses the trigger point, the RSU initiates a control cycle. During each control cycle, a CAV group is formed by including the on-ramp vehicle that has passed the trigger point, the subsequent on-ramp vehicles, and nearby mainline vehicles within a defined distance range. This CAV group then follows the controller's instructions to adjust speed, change lanes, and execute merging maneuvers. After completing these commands, these CAVs return to the car-following mode. Note that this study makes two assumptions. First, a fully cooperative environment is assumed, where all CAVs comply with centrally coordinated decisions. Second, the RSU is assumed to function as a perfect sensor, without communication delays or signal losses.

Controlling multiple CAV streams at a multi-lane freeway merging section includes three tasks: lane selection, vehicle sequencing, and trajectory planning. Here, lane selection and vehicle sequencing determine the target lanes and passing orders for CAVs driving through the merging section. Trajectory planning generates speed and acceleration profiles from their initial positions to the end of the merging section, ensuring CAVs reach selected lanes, follow assigned sequences, and maintain safety throughout the process. These tasks are interrelated: lane selection and vehicle sequencing impact trajectory planning, while trajectory costs, in turn, influence the determination of optimal lane selection and vehicle sequences.

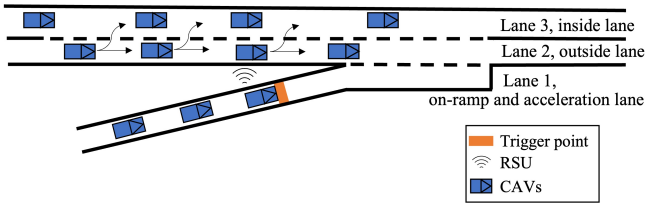


Fig. 1. The considered multi-lane freeway merging section.

TABLE I
Set, Parameter, and Variable Definitions

Sets	
I	set of all CAVs, $I = \{1, 2, \ldots, \lvert I\rvert\}$; $i, j \in I$.
L	set of all lanes, $L = \{l_1, l_2, l_3\}$; $l \in L$.
$\mathbb{O}(p_{i,t})$	set of occupied geometric space of vehicle i at time t, modeled as a rectangle centered at longitudinal position $p_{i,t}$.
$\mathcal{U}$	set of the admissible ranges for control inputs.
$\mathcal{V}$	set of the admissible range for vehicle speeds.
$\mathcal{P}$	set of the admissible ranges for vehicle positions.
Parameters	
$p_{i,\text{init}}$	initial position of vehicle $i \in I$.
$v_{i,\text{init}}$	initial speed of vehicle $i \in I$.
p_{final}	end position of the acceleration lane.
h	minimum time headway.
Variables	
$\gamma_{i,l}$	binary, equals 1 if CAV $i \in I$ is assigned to target lane $l \in L \backslash \{l_1\}$, 0 otherwise.
$\delta_{i,j}$	binary, equals 1 if CAVs $i \in I$ and $j \in I$ $(i \neq j)$ are assigned to the same target lane, 0 otherwise.
$\alpha_{i,j}$	binary, equals 1 if CAV $i \in I$ is assigned ahead of CAV $j \in I$ $(i \neq j)$ in the same lane, 0 otherwise.
T_i	continuous, the total travel time of CAV $i \in I$ through the merging section.
$p_i(t)$	continuous, the longitudinal position of CAV $i \in I$ at time t.
$v_i(t)$	continuous, the longitudinal velocity of CAV $i \in I$ at time t.
$u_i(t)$	continuous, the control input of CAV $i \in I$ at time t.

We first introduce the notations in Table I and then present a general mathematical model. Let $L = \{l_1, l_2, l_3\}$ denote the set of lanes, corresponding to the on-ramp, outside lane, and inside lane, respectively. Let $I = \{1, 2, \cdots, |I|\}$ represent a group of CAVs. Then, we introduce the decision variables summarized in Table I. For each vehicle $i \in I$ at time t, let continuous decision variables $p_{i,t}$, $v_{i,t}$, and $u_{i,t}$ denote the longitudinal position, speed, and acceleration, respectively. Here, longitudinal positions refer to the positions along each lane. Let the binary decision variable $\gamma_{i,l}$ denote the *target mainline lane choice* of vehicle $i \in I$, selected from $L \setminus \{l_1\}$. Specifically, $\gamma_{i,l}$ equals one if vehicle i selects mainline lane l as its target lane, and equals zero otherwise. Variable $\delta_{i,j}$ indicates whether vehicles i and j choose to go the same target lane, and variable $\alpha_{i,j}$ defines their driving order when $\delta_{i,j} = 1$. Finally, variable T_i denotes the total travel time of vehicle $i \in I$ through the merging section, defined as the duration from when the RSU is triggered to when vehicle i passes the endpoint of acceleration lane. According

to the notations defined above, the mixed integer nonlinear programming (MINLP) model is formulated as follows:

$$\min \sum_{i \in I} T_i \tag{1.1}$$

subject to:

$$\sum_{l \in L \setminus \{l_1\}} \gamma_{i,l} = 1 \qquad \forall i \in I \tag{1.2}$$

$$\sum_{l \in L \setminus \{l_1\}} \gamma_{i,l} \cdot \gamma_{j,l} = \delta_{i,j} \qquad \forall i,j \in I (i \neq j) \tag{1.3}$$

$$\alpha_{i,j} + \alpha_{j,i} = \delta_{i,j} \qquad \forall i,j \in I (i \neq j) \tag{1.4}$$

$$p_{i,0} = p_{i,\text{init}} \qquad \forall i \in I \tag{1.5}$$

$$v_{i,0} = v_{i,\text{init}} \qquad \forall i \in I \tag{1.6}$$

$$p_{i,T_i} = p_{\text{final}} \qquad \forall i \in I \tag{1.7}$$

$$\delta_{i,j}[(\alpha_{i,j} - \alpha_{j,i})(T_i - T_j) - h] >= 0 \qquad \forall i,j \in I (i \neq j) \tag{1.8}$$

$$\frac{dv_i}{dt}(t) = u_i(t) \qquad \forall i \in I, t \leq T_i \tag{1.9}$$

$$\frac{dp_i}{dt}(t) = v_i(t) \qquad \forall i \in I, t \leq T_i \tag{1.10}$$

$$\mathbb{O}(p_{i,t}) \cap \mathbb{O}(p_{j,t}) = \varnothing \qquad \forall i,j \in I (i \neq j), t \leq \min(T_i, T_j) \tag{1.11}$$

$$\gamma_{i,l} \in \{0,1\} \qquad \forall i \in I, l \in L \setminus \{l_1\} \tag{1.12}$$

$$\delta_{i,j} \in \{0,1\} \qquad \forall i,j \in I (i \neq j) \tag{1.13}$$

$$\alpha_{i,j} \in \{0,1\} \qquad \forall i,j \in I (i \neq j) \tag{1.14}$$

$$u_i(t) \in \mathcal{U} \qquad \forall i \in I, t \leq T_i \tag{1.15}$$

$$v_i(t) \in \mathcal{V} \qquad \forall i \in I, t \leq T_i \tag{1.16}$$

$$p_i(t) \in \mathcal{P} \qquad \forall i \in I, t \leq T_i. \tag{1.17}$$

1) Objective function: The objective is to minimize the total travel time of CAVs driving through a multi-lane freeway merging section.

2) Multi-vehicle combinatorial constraints: Constraints 1.2–1.4 ensure the orderly traffic flow distribution on the mainline lanes. Specifically, constraints 1.2 require that the target lanes for CAVs be selected from the mainline lanes, which means that each CAV can choose either outside lane (lane 2) or inside lane (lane 3). Note that, the target lane choice variable $\gamma_{i,l}$, l is defined only for $l \in L \setminus \{l_1\}$, thereby excluding the on-ramp and acceleration lane (lane 1) from being selected as a target lane. Constraints 1.3 state that $\delta_{i,j}$ equals 1 if and only if vehicles i and j choose the same target lane, and 0 otherwise. Then, 1.4 indicate that if two CAVs select the same target lane, they establish a unique driving sequence.

Additionally, constraints 1.12–1.14 specify that the associated variables as binary variables.

3) Trajectory Constraints: Constraints 1.5–1.11 are related to the vehicle's motion from its initial to the final state, ensuring that merging and lane-changing manoeuvres are completed. Constraints 1.5 and 1.6 specify the initial position and speed condition. Constraints 1.7 and 1.8 define the terminal conditions. Specifically, constraints 1.7 require that the terminal position is at the end of the merging section, p_{final}.

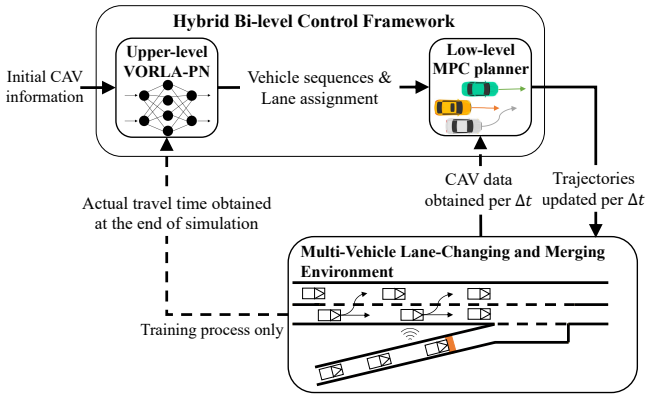


Fig. 2. The proposed hybrid bi-level control framework.

Constraints 1.8 ensure that each CAV's terminal passage time aligns with its lane selection and vehicle sequence decisions, while also maintaining a minimum time headway h between consecutive CAVs in the same target lane. Constraints 1.11 prevent collisions at any time step, where $\mathbb{O}(p_{i,t}) \in \mathbb{R}^2$ represents the geometric space occupied by vehicle i, modeled as a rectangle centered at the longitudinal position of vehicle i along its current lane at time t. the space occupied by vehicle i at time t. Constraints 1.9 and 1.10 describe the CAV's kinematic equations. Finally, constraints 1.15–1.17 define the permissible ranges for control inputs, speeds, and positions, respectively.

In summary, the on-ramp merging task of CAVs involves strategic decisions for vehicle sequences and lane selection to minimize total travel time, along with multi-vehicle trajectory planning that ensures safe movements while adhering to specified target lanes and vehicle sequences.

IV. Bi-level Control Framework

To solve the MINLP model (1.1)–(1.17) in real-time, our method adopts a hybrid bi-level control framework, as illustrated in Fig. 2. The upper level introduces a novel **V**ehicle **Or**dering and **L**ane **A**ssignment **P**olicy **N**etwork (VORLA-PN), which determines the vehicle sequence and lane assignment based on initial microscopic vehicle data (i.e., lane ID, position, and speed). The lower level develops an MPC planner to generate collision-free trajectories that implement the decisions made by the upper level.

We use a DRL algorithm to train VORLA, taking each simulation as a training instance. During each simulation, VORLA first outputs the vehicle sequence and lane assignment, which are passed to the MPC planner. The MPC planner then generates velocity commands for all vehicles at regular intervals (e.g., 200 ms) until they all pass through the merging section. The resulting travel time is provided as feedback to VORLA, serving as reward signals for updating the neural network parameters via the policy gradient formula. This framework effectively integrates the rapid inference ability of deep neural networks with the safety assurance of mathematical models. The following sections detail the upper-level VORLA policy network and the lower-level MPC planner.

V. VORLA Policy Network

A. Markov Decision Process Formulation

We formulate the problem of determining the vehicle driving sequence and lane assignment as an $|I|$-step Markov Decision Process (MDP), denoted as a tuple $\langle I, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} \rangle$, where I denotes the set of CAVs, $\mathcal{S}$ the state space, $\mathcal{A}$ the action space, $\mathcal{R}$ the reward function, and $\mathcal{T}$ the state transition function. The key MDP elements are defined as follows:

State: The state at each decision step k, defined as $s_k = (\mathbf{X}, \mathbf{N}_k) \in \mathcal{S}$, comprises two components: the initial CAV state $\mathbf{X}$ and the dynamic lane assignment state $\mathbf{N}_k$. $\boldsymbol{X} = \{X^{(i)}\}_{i=1}^{|I|}$ contains the information of all CAVs when the trigger point is activated, where $X^{(i)} = [p^{(i)}, v^{(i)}, \ell^{(i)}]^\mathsf{T}$ captures the longitudinal position, velocity, and lane ID of CAV $i \in I$. The lane assignment state $\boldsymbol{N}_k = [N_k^{l_2}, N_k^{l_3}]$ records the allocation of vehicles to mainline lanes l_2 and l_3 at step k. Specifically, both $N_k^{l_2}$ and $N_k^{l_3}$ are one-hot column vectors of length $|I|$, containing all zeros except for one single cell. This single cell, set to one, uniquely identifies the most recently assigned vehicle to the corresponding mainline lane at this step. For instance, if CAV i was assigned to lane 2 at step $k-1$, then $N_k^{l_2}$ would have its i^{th} cell set to 1, while the rest are set to zero.

Action: The action at step k is represented as $a_k = (c_k, l_k) \in \mathcal{A}$, where $c_k \in C_k$ selects one CAV from the set of available CAVs and $l_k \in L_k \backslash \{l_1\}$ selects a target lane from the set of permissible target mainline lanes. That is, at each decision step, only one CAV is chosen and assigned to a target lane, with this process continuing until all CAVs have been assigned. It is important to note that the sequence in which the CAVs are selected establishes the "vehicle sequence".

The available CAV set C_k is determined by the following two masking rules:

- Each CAV can be assigned to a target mainline lane once and only once.
- For CAVs initially in the same lane, vehicle selection starts with downstream CAVs and proceeds upstream.

The permissible lane set L_k depends on the initial lane ID of the selected vehicle. Mainline vehicles can choose a target lane from two mainline lanes (i.e., lanes l_1 and l_2), whereas on-ramp vehicles are restricted to the outside lane (i.e., lane l_2). Then, the corresponding masks $M_k^C \in \mathbb{R}^{|I|}$ and $\boldsymbol{M}_k^L \in \mathbb{R}^{2\times|I|}$ are constructed based on the rules of C_k and L_k to prevent invalid actions.

Reward: The objective is to minimize the total travel time for a group of CAVs. Accordingly, the sparse termination reward at the final decision step is denoted by $R_{|I|} = -\sum_{i\in I} T_i$, representing the negative of the objective function (1.1). The reward function is written as:

$$R_k = \begin{cases} 0, & k \in \{1, \ldots, |I|-1\}, \\ -\sum_{i\in I} T_i, & k = |I| \end{cases} \tag{2}$$

State Transition: Executing action $a_k = (c_k, l_k)$ on state s_k results in the next state s_{k+1} based on $\mathcal{T}(s_{k+1}|s_k, a_k)$. Specifically, given the action (c_k, l_k), the lane assignment state $\boldsymbol{N}_k = [N_k^{l_2}, N_k^{l_3}]$ is updated as follows:

$$N_{k+1}^l = \begin{cases} e_{c_k}, & l = l_k \\ N_k^l, & l \neq l_k \end{cases}, \tag{3}$$

where e_{c_k} is a one-hot vector with 1 at the c_k^{th} position and 0s elsewhere. At each step, one column is updated because only a single CAV is selected and assigned to one of the mainline lanes. If all CAVs have been assigned, the next state is the terminal state.

B. Sequential Decision-Making Process

Given initial CAV states $\mathbf{X}$, the policy network VORLA approximates a stochastic policy π_θ, which outputs probability distributions of a solution $\boldsymbol{a}$. A complete solution, $\boldsymbol{a} = (a_1, \ldots, a_{|I|})$, compromises a series of actions $a_k = (c_k, l_k)$ at each step k. Note that c_k represents the selection of a vehicle, and l_k denotes the assignment of that vehicle to a specific lane. The resulting sequence $(c_1, \ldots, c_{|I|})$ forms the driving sequence for $|I|$ CAVs. The process of generating a complete $\boldsymbol{a}$ can be factorized into a chain of conditional probabilities:

$$\pi_\theta(\boldsymbol{a}|\mathbf{X}) = \prod_{k=1}^{|I|} \pi_\theta(a_k|\boldsymbol{N}_k, \mathbf{X}) \tag{4}$$

This factorization allows the solution to be constructed incrementally, with the decision at each step a_k depending on the current state $\boldsymbol{N}_k$ and $\mathbf{X}$, where $\boldsymbol{N}_k$ depends on the previous actions, $a_{1:k-1}$. Consequently, a sequential decision-making process is developed to generate each partial solution a_k iteratively, ultimately building the complete solution $\boldsymbol{a}$.

The proposed VORLA network comprises a policy network π_θ and a baseline value network b_ϕ. The policy network π_θ includes an encoder that generates feature representation for CAVs and a decoder that produces a series of vehicle-lane probability matrices using (11). The action sequence $\boldsymbol{a}$ is determined by sampling from these probability matrices. The baseline value network b_ϕ is employed to reduce variance and facilitate learning efficiency. Fig 3 illustrates the VORLA network architecture, which is explained as follows.

1) Feature Encoder: We adopt the standard Transformer encoder [37] for feature extraction, as detailed in Appendix A. This encoder takes in the initial CAV states $\mathbf{X} \in \mathbb{R}^{|I|\times 3}$ and attends to information from all CAVs. The resulting vehicle state embedding $\boldsymbol{H}_\text{v}$ is represented as:

$$\boldsymbol{H}_\text{v} \in \mathbb{R}^{|I|\times d} = \text{TransformerEnc}\left(\boldsymbol{X}\right), \tag{5}$$

where d denotes the embedding dimensions.

2) Autoregressive Decoding: During the decoding process, the decoder is repeatedly executed, with each step generating a joint probability distribution of target CAVs and mainline lanes given the CAVs' embedding $\boldsymbol{H}_\text{v}$ and the lane assignment states $\boldsymbol{N}_k$. The resulting distribution is used to select the next CAV and its target lane.

The decoding process begins with lane feature extraction, using the embeddings of CAVs currently assigned to mainline

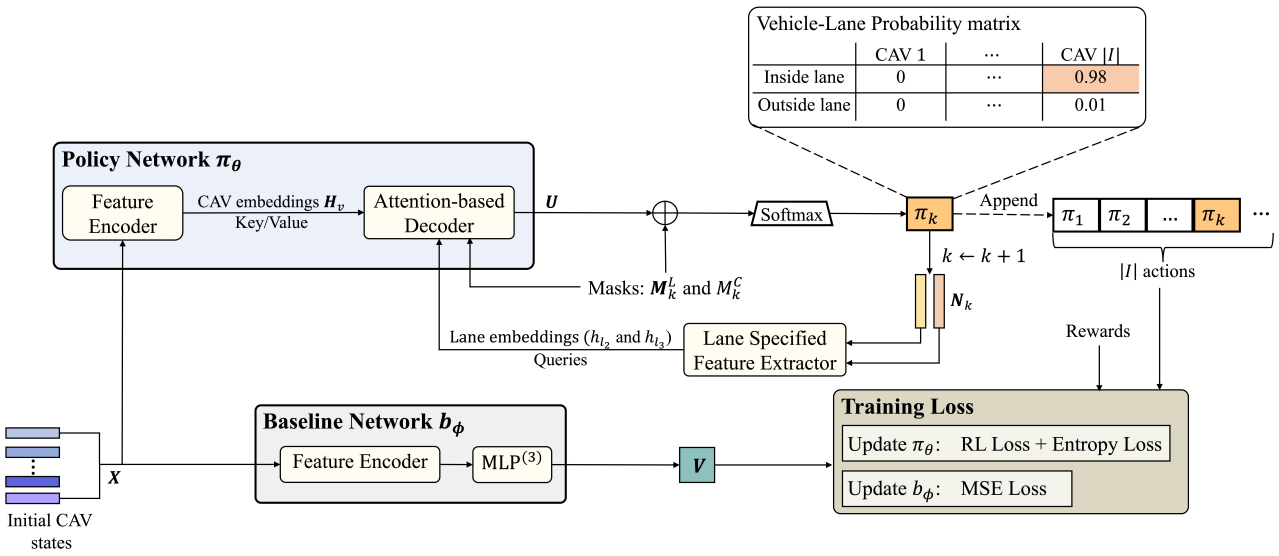


Fig. 3. The architecture of proposed VORLA network, which incorporates a encoder-decoder crafted policy network and a baseline value network. The autoregressive decoder constructs vehicle sequences and lane selections incrementally.

lanes as mainline lane features (*Step 1*). Next, the attention-based decoder computes a correlation matrix between the features of candidate vehicles and mainline lanes (*Step 2*). The decoder then generates a probability matrix based on this correlation, and samples it to select a vehicle and a mainline lane for the current step's decision. If there are still unassigned vehicles, the process loops back to Step 1 for the next decision step (*Step 3*).

Step 1. Lane feature extractor: Given the lane assignment states $\boldsymbol{N}_k$, the two feature vectors h_l, $l \in \{l_2, l_3\}$, are selected from the encoder output $\boldsymbol{H}_\text{v}$ as follows:

$$h_l \in \mathbb{R}^d = (N_k^l)^\mathsf{T} \cdot \boldsymbol{H}_\text{v}, \quad l \in \{l_2, l_3\} \tag{6}$$

which means the state embedding of CAVs currently assigned to the mainline lanes serves as the corresponding lane state. When no CAVs have been assigned to a lane, h_l is filled with trainable parameters.

Step 2. Attention-based decoder: Next, we learn the correlation between the features of current mainline lanes and candidate CAVs. We input the mainline lane features h_l as the query and the candidate CAV features as the key and value in the attention-based decoder.

The decoder comprises three layers of multi-head attention (MHA) modules. The MHA module is detailed in Appendix B.

The first MHA layer encodes correlation between each mainline lane and candidate CAVs, producing h'_l as follows:

$$h'_l \in \mathbb{R}^d = \text{MHA}(h_l, \boldsymbol{H}_\text{v} \odot M_k^C), \quad l \in \{l_2, l_3\}. \tag{7}$$

where h_l denotes mainline lane features, and the candidate CAV features are obtained through the mask operation $\boldsymbol{H}_\text{v} \odot M_k^C$. The symbol $\odot$ represents the element-wise product, and the mask $M_k^C \in \mathbb{R}^{|I|}$ are broadcast across each column of $\boldsymbol{H}_\text{v}$ to extract the permissible CAVs' features. h'_{l_2} and h'_{l_3} are processed separately, each through a dedicated MHA layer.

Next, we concatenate h'_{l_2} and h'_{l_3} into $h_\text{c} \in \mathbb{R}^{2d}$ and pass it through the second MHA layer to produce h_ls, a fused embedding integrating the features of the two mainline lanes.

$$h_\text{ls} \in \mathbb{R}^{2d} = \text{MHA}(h_\text{c}, h_\text{c}), \tag{8}$$

where $h_\text{c} \in \mathbb{R}^{2d} = \text{Concat}(h'_{l_2}, h'_{l_3})$.

After that, the third MHA layer computes the final correlation matrix $\boldsymbol{U} \in \mathbb{R}^{2\times|I|} = \text{Concat}(U_{l_2}, U_{l_3})$. The elements of $\boldsymbol{U}$ represent the correlation between the feature vector of a candidate CAV and that of a mainline lane, computed via a dot product operation. These correlation values are then transformed into probabilities, indicating the likelihood of selecting each candidate vehicle for each mainline lane. The correlation is calculated as follows:

$$U_l \in \mathbb{R}^{|I|} = C \cdot \tanh\left(\frac{h_\text{ls}^{(l)}(\boldsymbol{H}_\text{v} \odot M_k^C)^\mathsf{T}}{\sqrt{d}}\right), \quad l \in \{l_2, l_3\} \tag{9}$$

where $h_\text{ls}^{(l_2)}$ and $h_\text{ls}^{(l_3)}$ are d-dimensional vectors obtained by splitting h_ls. The tanh function clips the result within $[-C, C]$ ($C = 10$), following [40]. M_k^C is used to filter the non-permissible candidate CAVs according to the masking rules explained in Section V-A. Also, the non-permissible lanes for each CAV are filtered out using the mask $\boldsymbol{M}_k^L$ as follows,

$$u_{i,l} \in \mathbb{R} = \begin{cases} u_{i,l} & \text{if } m_i^l = 1 \text{ and } m_i^c = 1 \\ -\infty & \text{otherwise.} \end{cases} \quad i \in I, l \in \{l_2, l_3\} \tag{10}$$

where $u_{l,i}$, m_i^l, and m_i^c denote the elements of $\boldsymbol{U}$, $\boldsymbol{M}_k^L$, and M_k^C, respectively.

Step 3. Probability matrix generation: We interpret these correlation values as unnormalized log-probabilities (logits) and compute the final probability matrix $\boldsymbol{P}$ using a softmax function. Each element $p_{l,i}$ of $\boldsymbol{P}$ is calculated as follows:

$$p_{l,i} = \frac{e^{u_{l,i}}}{\sum_{i\in I, l\in\{l_2,l_3\}} e^{u_{l,i}}}. \tag{11}$$

Hence, the next passing CAV and the associated lane selection are determined by $\boldsymbol{P} \in \mathbb{R}^{2\times|I|}$.

In sum, the decoder uses the dependencies between lane situations and candidate CAV features to produce a selection probability matrix. During the training phase, actions are sampled from this multinomial probability distributions; during the inference phase, actions are selected greedily.

C. Leader-and-Lane-Specific Credit Assignment

During training, it is important yet challenging to correctly attribute the sparse termination cost, i.e. the total delay, to a sequence of actions (i.e., vehicle and lane selections). This paper proposes a **leader-and-lane-specific** credit assignment method to compute return G at each step, expressed as:

$$G(s_k, \underbrace{c_k, l_k}_{a_k}) = -\left[T(c_k) + \sum_{k'=k}^{|I|} T(c_{k'}) \cdot \mathbb{I}(l_{k'} = l_k)\right], \tag{12}$$

where action a_k contains two terms: c_k and l_k, which respectively denote the selected vehicle and target mainline lane at step k; $T(c_k)$ represents the travel time of the selected vehicle; $\mathbb{I}(l_{k'} = l_k)$ is an indicator function that equals 1 solely when $l_{k'}$ equals l_k. Equation (12) means that the return G only sums the travel time cost of the selected CAV and subsequent CAVs assigned to the same lane, rather than summing the costs of all subsequent CAVs.

This rule associates an action only with the rewards obtained afterwards, as prior rewards have no bearing on how good the action is. Moreover, the rule is consistent with microscopic traffic flow models, in which the leader's driving behaviour can affect their followers, but the followers do not influence the leader. Furthermore, this rule differentiates the mutual interactions of vehicles from different lanes. For example, lane-changing CAVs would impact their following vehicles in the target lane; otherwise, vehicles in different mainline lanes have no interaction. Additionally, the assignment of on-ramp CAVs has a more direct impact on outside-lane vehicles than inside-lane vehicles.

D. Policy Optimization

The REINFORCE with Baseline algorithm [38] is utilized to update the policy network π_θ, with a learnable baseline value network b_ϕ to reduce the variance of gradient estimates. The REINFORCE loss, $\mathcal{L}_{RL}$, is formulated as follows:

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{\tau\sim\pi_\theta}\left[(G(\tau) - b_\phi(\boldsymbol{X})) \cdot \log \pi_\theta(\tau)\right], \tag{13}$$

where $\mathbb{E}_{\tau\sim\pi_\theta}$ denotes the expectation over trajectories $\tau = (s_0, a_0, s_1, a_1, \dots)$, i.e., the sequences of states and actions, sampled from π_θ; G refers to the return, which is calculated based on (12); $b_\phi(\boldsymbol{X})$ estimates a expected return for each instance by the baseline network.

To discourage premature convergence, a negative entropy loss is adopted as follows:

$$\mathcal{L}_{\text{entropy}} = -\text{Entropy}(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{s,a\in\tau} \pi_\theta(a|s)\log(\pi_\theta(a|s))\right] \tag{14}$$

Consequently, the total loss for the policy network π_θ, denoted as $\mathcal{L}_\pi$, combines both the REINFORCE loss and the entropy loss:

$$\mathcal{L}_\pi = \mathcal{L}_{\text{RL}} + c_1 \cdot \mathcal{L}_{\text{entropy}}, \tag{15}$$

where c_1 is the coefficient of entropy loss. By minimizing $\mathcal{L}_\pi$, the policy distribution π_θ is optimized towards minimizing total travel delay. Regarding the baseline network b_ϕ, its loss is defined as follows:

$$\mathcal{L}_b = \mathbb{E}_{\tau\sim\pi_\theta}[(G(\tau) - b_\phi(\boldsymbol{X}))^2], \tag{16}$$

which aims to minimize the mean squared error (MSE) between the unbiased returns from the environment $G(\tau)$ and the estimated baseline values $b_\phi(\boldsymbol{X})$.

Lastly, Algorithm 1 outlines the training procedure.

Algorithm 1 Training process for our VORLA network

Input: Training dataset D, batch size B, number of epochs E, steps per epoch $|I|$.
Output: Network parameters θ.
for epoch $e = 1, E$ **do**
  Sample B instances from dataset D.
  for instance $i = 1, B$ **do**
    Set initial state $\boldsymbol{N}_1$ and $\boldsymbol{X}$ for instance i.
    for step $k = 1, |I|$ **do**
      Obtain a probability matrix $\pi_k \leftarrow \pi_\theta(a_k|\boldsymbol{N}_k, \boldsymbol{X})$.
      Sample an action $a_k \sim \pi_k$.
      Transit the next state $\boldsymbol{N}_{k+1}$.
    end for
    Execute the lower-level NMPC to implement target driving sequences and lane assignments $(\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{|I|})$ and to obtain vehicle delays $(r_1, \cdots, r_{|I|})$.
    Calculate returns $(G_1, \cdots, G_{|I|})$ based on (12).
    Obtain estimated baseline values $b_\phi(\mathbf{X})$.
  end for
  Calculate $\mathcal{L}_\pi$ based on (15).
  Calculate $\mathcal{L}_b$ based on (16).
  $\theta \leftarrow \text{Adam}(\theta, \nabla\mathcal{L}_\pi)$
  $\phi \leftarrow \text{Adam}(\phi, \nabla\mathcal{L}_b)$
end for
return policy network parameters θ.

VI. Lower-level Model Predictive Planner

We formulate the problem of short-horizon trajectory generation as an unconstrained optimization problem, which is solved by the open-source library LBFGS-Lite [41]. The objective is to minimize the following cost function:

$$\min_{U_i, V_i, P_i} [J_e, J_l, J_a, J_i, J_k, J_f] \cdot \lambda, \tag{17}$$

where $U_i = \{u_{i,t}\}_{t=0}^{N-1}$, $V_i = \{v_{i,t}\}_{t=0}^{N-1}$, and $P_i = \{p_{i,t}\}_{t=0}^{N-1}$ describe the trajectory of CAV i over a prediction horizon of N time steps. The time interval between steps is set to 0.2 second; λ is the weight vector used to trade off each cost term. The weight vector λ is heuristically tuned after

normalization of all cost terms, ensuring that the reciprocal avoidance, initial, kinematic, and feasibility conditions are all satisfied. To improve real-time performance, the MPC solver is warm-started with an initial solution that is dynamically consistent and near-feasible, allowing faster convergence.

1) Traffic Efficiency J_e: To optimize traffic efficiency, we minimize the cumulative difference between each vehicle's speed and the desired speed over the entire time horizon N, as defined below.

$$J_e = \sum_{k=0}^{N-1} (v_{i,k} - \bar{v})^2, \tag{18}$$

where $\bar{v}$ refers to the maximum free flow speed.

2) Target Sequence and Lane J_l: The upper level decides the target mainline lanes and vehicle passing orders, which consequently determines the target leading vehicle for each CAV. Hence, MPC must form the minimum spacing between any CAV i and its target leading vehicle:

$$J_l = \sum_{k=0}^{N-1} \psi_l(p_{i,k}, v_{i,k})^2 \tag{19}$$

where $\psi_l(p_{i,k}, v_{i,k}) = \min\{\hat{p}_k - p_{i,k} - v_{i,k}\cdot\tau - d_0, 0\}$ represents the penalty for insufficient spacing between CAV i and its target leader $\hat{p}$ at any time point. Here, insufficient spacing refers to a distance less than the product of the follower's speed and the minimum time gap τ, plus the standstill distance d_0.

3) Reciprocal Avoidance J_a: At any time, two consecutive CAVs in the same lane should maintain a minimum spacing. Therefore, the reciprocal avoidance penalty J_a is defined as:

$$J_a = \sum_{k=0}^{N-1} \psi_a(p_{i,k}, v_{i,k})^2 \tag{20}$$

where $\psi_a(p_{i,t}, v_{i,t}) = \min\{\tilde{p}_t - p_{i,t} - v_{i,t}\cdot\tau - d_0, 0\}$. Here, ψ_a is similar to ψ_l, but $\tilde{p}_t$ denotes the current leading vehicle, while $\hat{p}_t$ represents the target leading vehicle.

4) Initial Condition J_i: The starting position and speed of each CAV are fixed.

$$J_i = (p_{i,0} - p_{i,\text{init}})^2 + (v_{i,0} - v_{i,\text{init}})^2 \tag{21}$$

5) Kinematic Condition J_k: This condition links control inputs, velocities, and positions at two consecutive time points, as follows:

$$\begin{aligned} J_k = \sum_{k=0}^{N-2} \Big[&(v_{i,k+1} - v_{i,k} - u_{i,k}\Delta t)^2 + \\ &(p_{i,k+1} - p_{i,k} - \frac{v_{i,k} + v_{i,k+1}}{2} \cdot \Delta t)^2 \Big] \end{aligned} \tag{22}$$

where Δt denotes the time interval. Here, velocities are linearly dependent on the control input, and positions have a parabolic relationship with velocity.

TABLE II
VEHICLE PARAMETERS AND HYPERPARAMETERS FOR TRAINING DRLS

Vehicle parameter	Value	Hyper-parameter	Value
Minimum time headway h	1.2s	Batch size $\|B\|$	64
Jam distance d_0	10m	Learning rate η_θ	1e-4
Maximum speed $\bar{v}$	120km/h	Learning rate η_ϕ	1e-4
Maximum acceleration $\bar{a}$	2m/s^{-2}	Entropy weight c_1	5e-3
Maximum deceleration $\bar{b}$	4m/s^{-2}	Embedding dim d	128

6) Feasibility condition J_f: We limit the value of velocity and control input within feasible regions.

$$J_f = \sum_{k=0}^{N-1} \psi_v(v_{i,k}) + \psi_u(u_{i,k}) \tag{23}$$

where ψ_v and ψ_u are respectively defined as:

$$\psi_v = \begin{cases} (v - \bar{v})^2, & v > \bar{v} \\ 0, & v \in \mathcal{V} \\ (v - \underline{v})^2, & v < \underline{v} \end{cases} \qquad \psi_u = \begin{cases} (u - \bar{u})^2, & u > \bar{u} \\ 0, & u \in \mathcal{U} \\ (u - \underline{u})^2, & u < \underline{u} \end{cases} \tag{24}$$

VII. SIMULATION RESULTS

Sections VII-A and VII-B describe our simulation settings and training data generation. Sections VII-C, VII-D, and VII-E evaluate the efficacy of our approach, our credit assignment method, and learning techniques, respectively. Section VII-F presents a case study to illustrate the decision-making process and the generated trajectories.

A. Simulation Settings

The microscopic traffic simulator SUMO [39] is adopted to conduct microscopic simulation. The simulated freeway segment consists of two mainline lanes and one on-ramp, as depicted in Fig. 4. The mainline lanes extend for 1.7 kilometers, comprising three sections: the upstream segment, where CAVs are randomly generated; the cooperation segment, where merging and lane-changing behavior occur; and the downstream segment, which fully covers the affected areas.

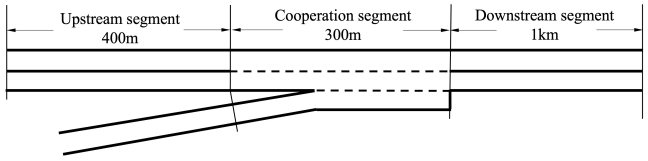


Fig. 4. Simulated road geometry.

Table II summarizes the values of vehicle parameters and the hyper-parameters used for training the DRL methods. The vehicle parameters include the minimum time headway, jam distance, speed range, and acceleration range. A minimum time headway of 1.2 seconds is adopted in this study. Compared to mainline car-following, merging maneuvers require more conservative spacing to account for uncertainties in vehicle acceleration, deceleration, and interactions between merging and mainline vehicles.

B. Training Configuration and Data Generation

The training-related hyper-parameters directly affect the training process and data volume. Specifically, the model is trained using a batch size of 64, and each roll-out involves $|I|$ vehicles and thus $|I|$ decision steps, yielding $64 \times |I|$ training samples per batch. As the training typically involves several thousand batches, the total training data volume reaches the scale of millions of samples. As for the data generation, two types of datasets are used in this study: one is synthetically generated, and the other is extracted from real-world dataset.

1) Synthetic Data Generation: Each episode randomly initializes 15 CAVs along the upstream segment, with 25–30% placed on the on-ramp, 35–40% on the outside lane, and 30–40% on the inside lane. The initial time headway between any two adjacent vehicles within the same lane is randomly drawn from a uniform distribution between 1.2 and 2.0 seconds. The initial speeds of mainline CAVs and on-ramp CAVs are sampled uniformly from 100–120 km/h and 80–104 km/h, respectively.

2) Real-world Dataset: We extract data from the Entries and Exits Drone (ExiD) real-world dataset recorded on German highways [42], which includes detailed on-ramp merging scenarios. A total of 1,784 merging cases are extracted, each capturing the position and speed of vehicles at the beginning of the merging segment. Among these, 80% of the cases are for training and 20% for testing.

C. Comparative Analysis

We compare our method with following methods:

- SA-10: Simulated Annealing (SA) is a metaheuristic method that uses a guided neighborhood search, commonly applied to discrete search spaces. Here, it is employed to search for vehicle orders and lane assignments. SA is executed with a 10-minute time limit (hence referred to as SA-10).
- BF-10: Brute Force (BF) search over all feasible discrete decisions, with a 10-minute time limit imposed (hence referred to as BF-10).
- FIFO: A rule-based method that first rotates CAVs from the ramp and outside lanes into a shared straight line, and then orders them given the first-in-first-out (FIFO) way.
- IPPO: Independent Proximal Policy Optimization (IPPO), where each CAV learns its own policy using ego-centric observations to optimize individual performance (e.g., ego speed and smoothness) [29], [43].
- MAPPO: Multi-Agent Proximal Policy Optimization (MAPPO), which leverages global observations of neighboring CAVs to learn policies that optimize collective performance (e.g., average speed) [30], [44].

1) Synthetic Data Validation: We begin evaluating our method on synthetic scenarios. Table III presents the statistics results of different methods on 400 instances. The evaluation metrics include travel delay and computation time. Our method demonstrates superior real-time performance compared to rule-based and metaheuristic approaches. In terms of solution quality, it achieves the lowest mean value of travel delay (66.8s), significantly outperforming FIFO (93.0s) and SA-10 (92.1s) and slightly better than BF-10 (69.8). In terms of computation time, our method requires only milliseconds for network inference. While rule-based methods are faster, they produce significantly lower-quality solutions. In contrast, other methods require tens of minutes, making them unsuitable for real-time applications.

TABLE III
PERFORMANCE COMPARISON

	VORLA (our)	FIFO	SA-10	BF-10
Travel delay (s)	**66.8** $\pm$ 12.9	93.0 $\pm$ 23.0	92.1 $\pm$ 23.0	69.8 $\pm$ 8.7
Computation time (s)	0.006	$\leq$ 0.001	600	600

2) Real-world Data Validation: We further evaluate the proposed method using real-world scenarios. Vehicles are initialized using the recorded positions and speeds, and the merging process is then simulated using the proposed control framework. Table IV presents the travel delay results on the test set from real-world dataset, comparing our method with rule-based, metaheuristic, and multi-agent reinforcement learning methods for cases with different numbers of vehicles (i.e., from 3 vehicles to 15 vehicles). The performance of the FIFO method decreases as the number of vehicles increases. SA-10 struggles to find good solutions using its guided random search. BF-10 can find high-quality solutions for cases with fewer vehicles by enumerating feasible options, but its effectiveness diminishes as the vehicle count grows. In contrast, our method achieves the lowest delay in cases with both small and large numbers of vehicles (3, 11, 13, and 15 vehicles). For moderate vehicle numbers (5, 7, and 9 vehicles), the average total delay is only 0.1–2 seconds longer than that of BF-10. This demonstrates that our method not only provides high-quality solutions in milliseconds but also scales effectively to varying numbers of vehicles. Also, compared with decentralized learning-based methods such as IPPO and MAPPO, which directly output the acceleration or deceleration of each CAV at every time step, our method employs a centralized control framework that abstracts decision-making into vehicle-lane assignments and ordering. This high-level coordination enables more structured planning and leads to consistently better performance.

D. Comparison of Credit Assignment Methods

To demonstrate the effectiveness of our proposed leader-and-lane-specific credit assignment mechanism, we compare it with other reward shaping approaches:

- Vehicle-specific reward: Each action is only associated with the negative travel delay of the selected vehicle.
- Terminal reward: An entire sequence of actions is associated with a sparse terminal reward, i.e., the negative total travel delay of a CAV group.

As illustrated in Fig. 5, under different credit assignment approaches, the policy network converges stably but achieves different performance. The proposed assignment approach significantly helps the policy network converge to solutions

TABLE IV
COMPARISON OF TRAVEL DELAY FOR VARYING NUMBERS OF VEHICLES

# of CAVs	Travel delay (mean value with standard deviation) (s)					
	VORLA (our)	FIFO	SA-10	BF-10	IPPO	MAPPO
3	$\mathbf{1.6} \pm 1.7$	2.1 ± 1.9	5.7 ± 6.9	$\mathbf{1.6} \pm 1.8$	2.4 ± 1.8	2.3 ± 1.7
5	2.9 ± 2.2	4.1 ± 3.4	8.8 ± 10.0	$\mathbf{2.5} \pm 1.9$	3.4 ± 2.1	3.3 ± 2.0
7	4.8 ± 3.2	7.0 ± 6.7	19.1 ± 14.1	$\mathbf{4.7} \pm 3.2$	5.7 ± 2.8	5.2 ± 3.1
9	7.7 ± 5.7	8.0 ± 5.8	30.5 ± 20.6	$\mathbf{5.7} \pm 3.9$	10.2 ± 6.6	9.5 ± 5.4
11	$\mathbf{16.7} \pm 11.5$	19.0 ± 11.9	46.5 ± 18.1	25.4 ± 10.8	17.9 ± 6.9	16.9 ± 5.4
13	$\mathbf{20.5} \pm 10.2$	23.3 ± 11.8	85.0 ± 13.3	35.3 ± 14.0	22.3 ± 10.1	20.8 ± 8.0
15	$\mathbf{27.3} \pm 8.6$	35.3 ± 16.2	79.9 ± 33.5	67.2 ± 20.8	34.7 ± 11.0	30.6 ± 7.4

with the lowest travel delay. It is noteworthy that the proposed approach explicitly associates each action with the delays of a selected vehicle and those subsequently affected. In contrast, the vehicle-specific approach associates each action solely with the delay of the selected vehicle. Although the explicitness of the vehicle-specific approach enables the network to converge stably to a local optimum, its performance is ultimately inferior to that of our assignment approach because it fails to account for the impact of actions on other vehicles. Meanwhile, the terminal reward approach associates each action with the total delay, meaning that each vehicle's assignment is responsible for the delays of all vehicles. However, this makes the network struggle to learn which vehicles are impacted by a given action and which are not.

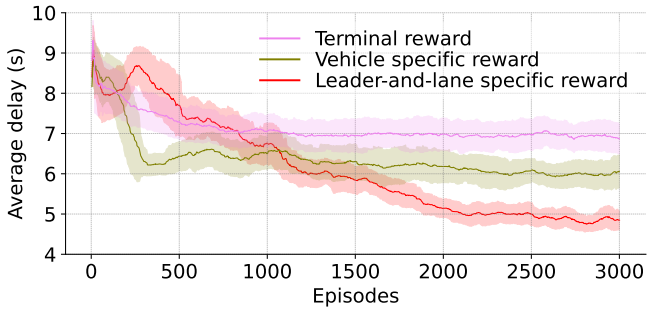


Fig. 5. Learning curves under different credit assignment approaches.

E. Ablation studies

Our DRL method incorporates learnable baseline and entropy regulation techniques. We conduct ablation studies to assess the effectiveness of these techniques. Fig. 6 displays the learning curves under three conditions: with baseline and entropy regulation, without the baseline, and without entropy.

The key observations are as follows: First, the learning curve aided by the two technologies reaches the lowest cost, demonstrating their effectiveness in converging to superior solutions. Second, in the absence of the baseline, variance increases, adversely affecting the learning process. Lastly, without entropy regulation, the learning curve exhibits a pronounced bend after 500 episodes, indicating rapid and premature convergence.

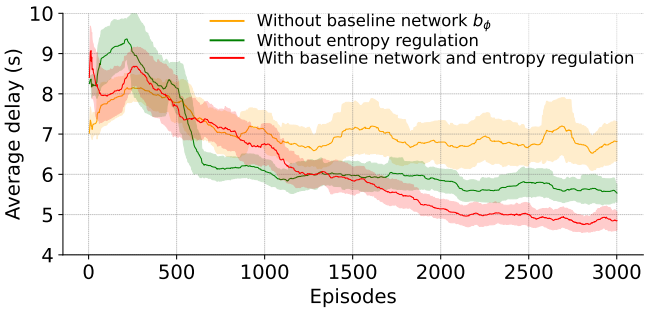


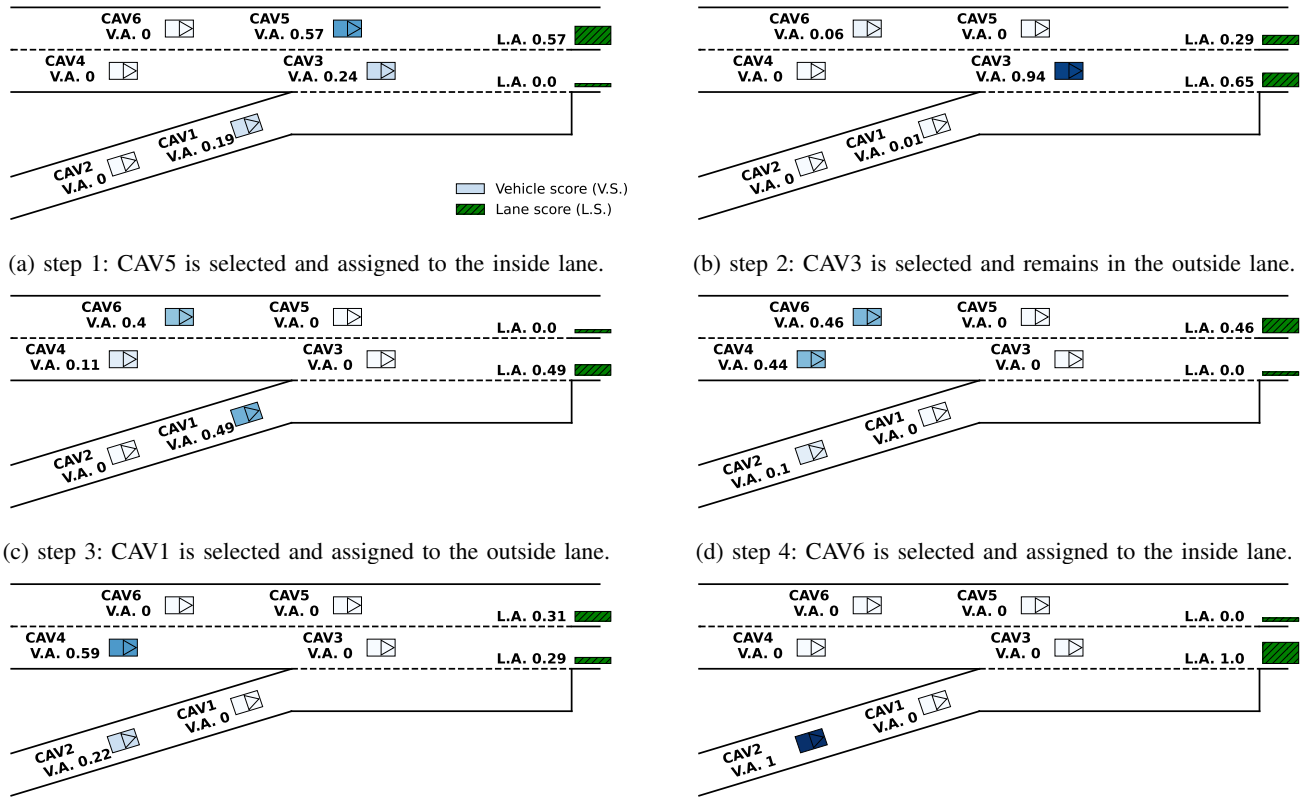
Fig. 6. Learning curves for the ablation study of baseline network and entropy regulation.

F. Case Study

To illustrate the decision-making process and the detailed trajectories generated by the proposed approach, a case study involving six CAVs is conducted. The initial positions and velocities of all CAVs are presented in Table 7.

	CAV1	CAV2	CAV3	CAV4	CAV5	CAV6
Speed (m/s)	23.3	22.9	27.0	26.5	30.0	29.2
Longitudinal position (m)	126.7	104.6	151.4	105.3	146.0	115.9

Fig. 7 visualizes the vehicle-lane probability matrices generated during each decision step. The values in the matrices represent the probability of selecting the next target lane and vehicle. The vehicle score (V.S.) represents the probability of selecting a vehicle, calculated as the sum of two lane scores (L.S.). Each lane score indicates the preference of the most likely selected vehicle for two mainline lanes. Higher V.S. and L.S. values correspond to a higher probability of choosing and assigning the vehicle to the lane. As shown in Fig. 7 (a), at the first step, CAV5 is the first selected vehicle and assigned to the inside lane. CAV3 has the second highest vehicle score, making it a strong candidate for the first selection. Unsurprisingly, at the second step, CAV3 is then selected and assigned to the outside lane, consistent with the decision made in the first step. At the third step, both CAV1 and CAV6 are strong candidates, compared to CAV4, due to their more downstream positions. The policy network selects CAV1 first, followed by CAV6 in the subsequent step. Finally, CAV4 and CAV2 are selected in order, with CAV4 chosen first because of its higher speed and more downstream



(a) step 1: CAV5 is selected and assigned to the inside lane.

(b) step 2: CAV3 is selected and remains in the outside lane.

(c) step 3: CAV1 is selected and assigned to the outside lane.

(d) step 4: CAV6 is selected and assigned to the inside lane.

(e) step 5: CAV4 is selected and changed to the inside lane.

(f) step 6: CAV2 is selected and assigned to the outside lane.

Fig. 7. Attention scores of vehicles and lanes at each decision step

position. Notably, CAV4 changes to the inside lane to avoid the interference from the two on-ramp vehicles, which is a preferred decision for optimizing overall traffic efficiency.

Fig. 8 illustrates the longitudinal and lateral position trajectories of these CAVs at different moments, obtained by solving the proposed method. Circled numbers in Fig. 8 mark time moments, with circled 1 indicating the starting moment. Also, Fig. 9 (a) depicts the speed profiles of on-ramp vehicles (CAV1 and CAV2) and the leading outside-lane vehicle (CAV3), while Fig. 9 (b) shows the speed profiles of lane-changing vehicle (CAV4) and inside-lane vehicles (CAV5 and CAV6).

VIII. Summary and Discussion

In this paper, we first mathematically express CAV-based multi-lane freeway merging control problems, clarifying the connections between lane selection, vehicle sequencing, and trajectory planning. We then propose a hybrid bi-level control approach that combines DRL's computational efficiency with MPC's safety guarantees to optimize vehicle sequences, lane selections, and trajectories under strict real-time requirements. Specifically, we frame vehicle sequencing and lane selection as a multi-step decision-making process and design an upper-level DRL agent to auto-regressively construct driving sequences and lane selections based on prior decisions. Our lower-level MPC planner replans trajectories at regular intervals to accomplish the upper-level instructions on lane assignments and right-of-way priorities. To effectively train our DRL agent, our new leader-and-lane-specific credit assignment mechanism associates specific vehicle delays with specific decisions, thereby distinguishing the mutual influence among vehicles based on their relative order and lane assignments. Simulation results show that under our hybrid approach, vehicle delays are significantly better than those of rule-based and learning-based methods and comparable to the results of metaheuristic search methods that require tens of minutes. It demonstrates our approach's superior millisecond-level real-time performance, effectively trading-off solution quality and computation time. Moreover, whether coordinating several vehicles or over a dozen, our approach consistently performs well, showing excellent scalability for real-world scenarios with varying vehicle numbers. Furthermore, simulation results show that our credit assignment method does enhance the DRL agent's ability to learn a better strategy.

Despite the promising results, this study has following limitations and suggests directions for future research. First, the proposed framework assumes fully cooperative vehicle behavior, whereas real-world traffic involves non-cooperative vehicles. In such cases, an additional safety emergency mechanism, e.g., a constant full-deceleration strategy, can serve as a fallback to enhance the practical viability. Moreover, extending the proposed framework to mixed-traffic conditions would further strengthen its applicability in reality. Second, the centralized control approach implicitly assumes cooperation among original equipment manufacturers, which is

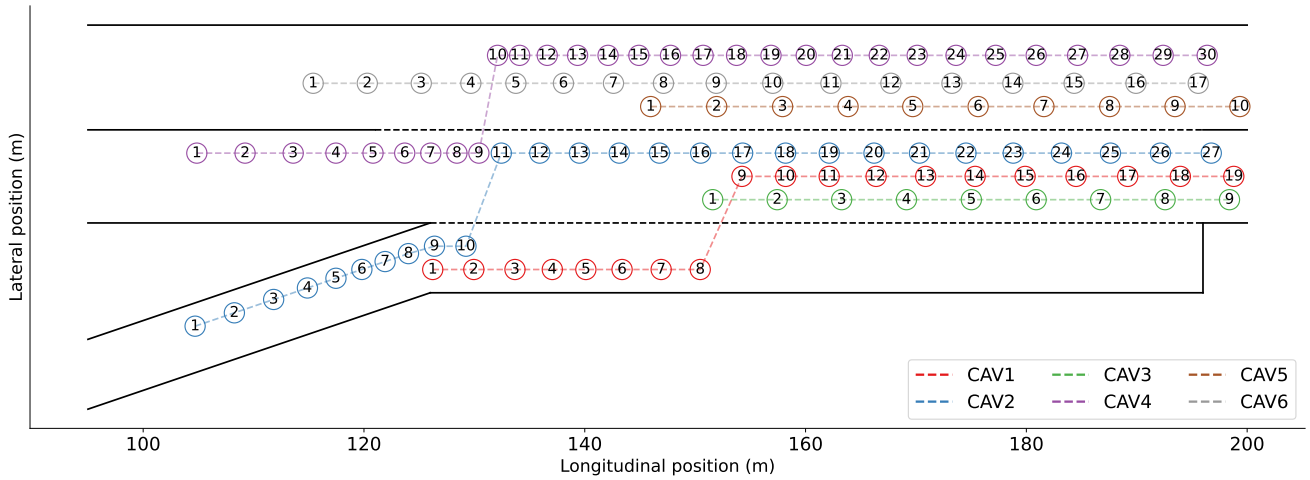
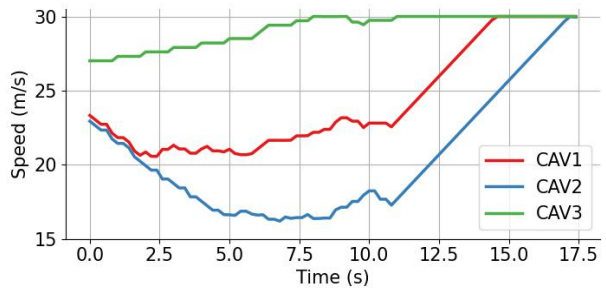
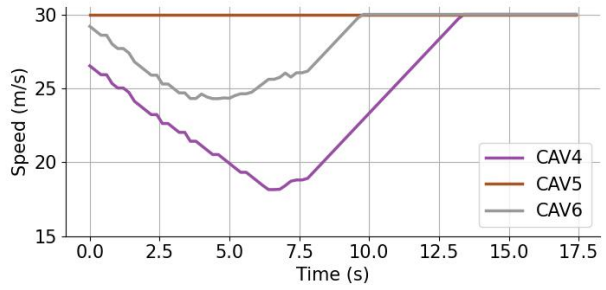


Fig. 8. Lateral and longitudinal positions of CAVs in the case study.

challenging due to commercial and standardization barriers. One promising direction is to adopt a decentralized DRL-based decision-making framework, which offers greater scalability and practicality under such constraints. Third, the RSU is assumed to be an ideal infrastructure component. In practice, communication delays, perception errors, and signal losses may affect system performance. Moreover, the time and financial costs associated with installing and maintaining such infrastructure need careful evaluation in future research. Lastly, the combinatorial nature of the problem remains a key obstacle to obtaining the optimal solution. When scheduling over twenty CAVs simultaneously in multi-lane scenarios, the extremely large search space makes it difficult to obtain an effective coordination strategy.



(a) Speed profiles of CAVs assigned to the outside lane.



(b) Speed profiles of CAVs assigned to the inside lane.

Fig. 9. Speed profiles of CAVs in the case study.

Appendix A
Transformer Encoder

The standard Transformer encoder [37] is adapted to take in a sequence of tokens and produce a representation embedding of the entire sequence. The encoder consists of N identical layers, each of which contains two components: a multi-head self-attention module and a followed position-wise fully connected network. Each component's output is wrapped with a residual connection layer, followed by a normalization layer, ensuring stable gradient flow.

Appendix B
Multi-head Attention Module

The multi-head attention (MHA) module [37] is a crucial component of our network. The MHA module takes in a query source, $h^q \in \mathbb{R}^d$, and a key-value source, $h^{k,v} \in \mathbb{R}^d$. Both h^q and $h^{k,v}$ are projected H times into different subspaces using linear layers, where H is denoted as the number of heads. For each head $h \in \{1, 2, \ldots, H\}$, the query, key, and value vectors are calculated as follows:

$$\mathbf{Q}_h = \mathbf{W}_h^Q h^q \tag{B.1}$$

$$\mathbf{K}_h = \mathbf{W}_h^K h^{k,v} \tag{B.2}$$

$$\mathbf{V}_h = \mathbf{W}_h^V h^{k,v}, \tag{B.3}$$

where h^q is the query source; $h^{k,v}$ is the key-value source; $\mathbf{W}_h^Q$, $\mathbf{W}_h^K$, and $\mathbf{W}_h^V \in \mathbb{R}^{d_h \times d}$ are learnable weight matrices, and $d_h = d/H$. Then, each attention-based head α_h is determined through the scaled-dot product operation:

$$\alpha_h = \text{Attention}(Q_h, K_h, V_h) = \text{softmax}\left(\frac{Q_h {K_h}^T}{\sqrt{d_h}}\right) V_h \tag{B.4}$$

The last operation of the MHA module is to concatenate all heads together:

$$\text{MHA}(h^q, h^{k,v}) = \text{Concat}(\alpha_1, \alpha_2, \cdots, \alpha_H)\boldsymbol{W}^O \tag{B.5}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{d \times d}$ is the learnable matrix for the output layer. Hence, the output of MHA aggregates the key/value, guided by the query.

REFERENCES

[1] Y., Han, and S. Ahn, "Stochastic modeling of breakdown at freeway merge bottleneck and traffic control method using connected automated vehicle," *Transp. Res. Part B Methodol.*, vol. 107, pp. 146–166, Dec. 2018.

[2] W.Y., Mergia, D., Eustace, D. Chimba, and M., Qumsiyeh, "Exploring factors contributing to injury severity at freeway merging and diverging locations in Ohio," *Accid. Anal. Prev.*, vol. 55, pp. 202–210, Mar. 2013.

[3] A., Srivastava, and N., Geroliminis, "Empirical observations of capacity drop in freeway merges with ramp control and integration in a first-order model," *Transp. Res. Part C Emerging Technol.*, vol. 30, pp. 161–177, Mar. 2013.

[4] C., Lee, B., Hellinga, and F., Saccomanno, "Evaluation of variable speed limits to improve traffic safety," *Transp. Res. Part C Emerging Technol.*, no. 3, pp. 213–228, Aug. 2006.

[5] H., Hadj-Salem, J. M., Blosseville, and M., Papageorgiou, "ALINEA: A local feedback control law for on-ramp metering," *Transp. Res. Rec.*, vol. 1320, pp. 58–67, Mar. 1991.

[6] M., Wang, "Infrastructure assisted adaptive driving to stabilise heterogeneous vehicle strings," *Transp. Res. Part C Emerging Technol.*, vol. 91, pp. 276–295, Apr. 2018.

[7] J., Shen, and L. Du, "Sequential feasibility and constraint properties of CAV platoons under various vehicle dynamics and safety distance constraints," *Transp. Res. Part B Methodol.*, vol. 185, pp. 102966, May 2024.

[8] Z., Zhong, E. E., Lee, M., Nejad, and J. Lee, "Influence of CAV clustering strategies on mixed traffic flow characteristics: an analysis of vehicle trajectory data," *Transp. Res. Part C Emerging Technol.*, vol. 115, pp. 102611, Mar. 2020.

[9] Y., Zhou, J., Chen, E. Chung, and K., Ozbay, "CAV-Enabled Active Resolving of Temporary Mainline Congestion Caused by Gap Creation for On-Ramp Merging Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6873–6888, Jul. 2023.

[10] J., Kim, D., Lim, Y., Seo, J. So, and H., Kim, "Influence of dedicated lanes for connected and automated vehicles on highway traffic flow," *IET Intel. Transport Syst.*, vol. 17, no. 4, pp. 678–690, Oct. 2023.

[11] M., Da Lio, A., Cherubini, G. P., Rosati Papini, and A., Plebe, "Complex self-driving behaviors emerging from affordance competition in layered control architectures," *Cogn. Syst. Res.*, vol. 1, no. 79, pp. 4–14, Jun. 2023.

[12] Y., Zhou, E., Chung, A. Bhaskar, and M.E., Cholette, "A state-constrained optimal control based trajectory planning strategy for cooperative freeway mainline facilitating and on-ramp merging maneuvers under congested traffic," *Transp. Res. Part C Emerging Technol.*, vol. 109, pp.321–342, Nov. 2019.

[13] Y., Zhou, M.E., Cholette, A. Bhaskar, and E., Chung, "Optimal vehicle trajectory planning with control constraints and recursive implementation for automated on-ramp merging," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 9, pp. 3409–3420, Sept. 2018.

[14] X., Hu, and J., Sun, "Trajectory optimization of connected and autonomous vehicles at a multilane freeway merging area," *Transp. Res. Part C Emerging Technol.*, vol. 101, pp. 111–125, Feb. 2019.

[15] M., Karimi, C., Roncoli, C., Alecsandru, and M., Papageorgiou, "Cooperative merging control via trajectory optimization in mixed vehicular traffic," *Transp. Res. Part C Emerging Technol.*, vol. 116, pp. 102663, May 2020.

[16] S., Jing, F., Hui, X., Zhao, J., Rios-Torres, and A. J., Khattak, "Integrated longitudinal and lateral hierarchical control of cooperative merging of connected and automated vehicles at on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24248–24262, Dec. 2022.

[17] H., Liu, W., Zhuang, G., Yin, Z., Li, and D. Cao, "Safety-critical and flexible cooperative on-ramp merging control of connected and automated vehicles in mixed traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 2920–2934, Mar. 2023.

[18] H., Pei, S., Feng, Y., Zhang, and D., Yao, "A cooperative driving strategy for merging at on-ramps based on dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11646–11656, Dec. 2019.

[19] Z., Tang, H., Zhu, X., Zhang, M., Iryo-Asano, and H., Nakamura, "A novel hierarchical cooperative merging control model of connected and automated vehicles featuring flexible merging positions in system optimization," *Transp. Res. Part C Emerging Technol.*, vol. 138, pp. 103650, Mar. 2022.

[20] J., Shi, K., Li, C., Chen, W., Kong, and Y., Luo, "Cooperative merging strategy in mixed traffic based on optimal final-state phase diagram with flexible highway merging points," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 11185–11197, Oct. 2023.

[21] J., Chen, Y., Zhou, and E., Chung, "An integrated approach to optimal merging sequence generation and trajectory planning of connected automated vehicles for freeway on-ramp merging sections," *IEEE Trans. Intell. Transp. Syst.*, Feb. 2023.

[22] Y., Xie, G., Lu, F., Zheng, P., Cao, and X, Liu, "A hierarchical approach for integrating merging sequencing and trajectory optimization for connected and automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, Jul. 2024.

[23] N., Chen, B., van Arem, T., Alkim, and M., Wang, "A hierarchical model-based optimization control approach for cooperative merging by connected automated vehicles," *IEEE Trans. on Intell. Transp. Sys.*, vol. 22, no. 12, pp.7712–7725, Jul. 2020.

[24] N., Chen, B., van Arem, and M., Wang, "Hierarchical optimal maneuver planning and trajectory control at on-ramps with multiple mainstream lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23 no. 10, pp. 18889–18902, Oct. 2022.

[25] R. A., Dollar, and A., Vahidi, "Multilane automated driving with optimal control and mixed-integer programming," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2561–2574, Nov. 2021.

[26] L., Yang, J., Zhan, W. L., Shang, S., Fang, G., Wu, X., Zhao, and M., Deveci, "Multi-lane coordinated control strategy of connected and automated vehicles for on-ramp merging area based on cooperative game," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13448–13461, Nov. 2023.

[27] H., Yu, H., Tseng, and R., Langari, "A human-like game theory-based controller for automatic lane changing," *Transp. Res. Part C Emerging Technol.*, vol. 88, vol.140–158, Feb. 2018.

[28] C., Wei, Y., He, H., Tian, and Y., Lv, "Game theoretic merging behavior control for autonomous vehicle at highway on-ramp," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21127–21136, Nov. 2022.

[29] D., Chen, M. R., Hajidavalloo, Z., Li, K., Chen, Y., Wang, L., Jiang, and Y., Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 11623–11638, Nov. 2023.

[30] J., Hu, X., Li, W., Hu, Q., Xu, and D., Kong, "A cooperative control methodology considering dynamic interaction for multiple connected and automated vehicles in the merging zone," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12669–12681, Sept. 2024.

[31] S., Hwang, K., Lee, H., Jeon, and D., Kum, "Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17594–17606, Oct. 2022.

[32] O., Vinyals, M., Fortunato, and N, Jaitly, "Pointer networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015.

[33] W., Kool, H., van Hoof, and M., Welling, "Attention, learn to solve routing problems," 2018. [Online]. Available: arXiv:1803.08475.

[34] X. Bi, R. Wang, H. Ye, Q. Hu, S. Bu and E. Chung, "Real-Time Scheduling of Electric Bus Flash Charging at Intermediate Stops: A Deep Reinforcement Learning Approach," *IEEE Trans. Transp. Electrif.*, vol. 10, no. 3, pp. 6309–6324, Sept. 2024.

[35] J., Zhang, S., Li, and L., Li, "Coordinating CAV swarms at intersections with a deep learning model, *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6280–6291, Jun. 2023.

[36] C., Jiang, H., Liu, C., Qiu, S., Zhang, and W., Zhuang, "Ramp merging sequence and trajectory optimization for connected and autonomous vehicles using deep reinforcement learning," in *Proc. IEEE 18th Int. Confer. Adv. Motion Control (AMC)*, Japan, 2024, pp. 1–7.

[37] A., Vaswani, et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, USA, 2017, pp. 6000–6010.

[38] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, May 1992.

[39] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, Spain, 2011.

[40] I., Bello, H., Pham, Q. V., Le, M., Norouzi, and S, Bengio, "Neural combinatorial optimization with reinforcement learning," 2016. [Online]. Available: arXiv:1611.09940.

[41] D. C., Liu, and J., Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, pp. 503–528, Aug. 1989.

[42] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exiD dataset: a real-world trajectory dataset of highly interactive highway scenarios in Germany", in *Proc. IEEE Intell. Veh. Symp.*, Germany, 2022, pp. 958–964.

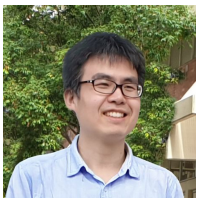
[43] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in

the StarCraft multi-agent challenge?," arXiv preprint arXiv:2011.09533, 2020.

[44] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games", in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24611–24624, 2022.

Jieming Chen is currently a postdoctoral fellow in the Department of Electrical and Electronic Engineering at the Hong Kong Polytechnic University. He obtained his Ph.D. degree from the same department in 2025. He also holds a B.Eng. degree in Electrical Engineering from Shanghai Maritime University in 2017 and an M.S. degree in Control Engineering from the Technical University of Kaiserslautern in 2021. His research focuses on intelligent transportation systems.

Yifeng Zhang received the B.Eng. degree in Vehicle Engineering from Hunan University (HNU), Changsha, China, in 2020, the M.Sc. degree in Mechanical Engineering from the National University of Singapore (NUS), Singapore, in 2021, and the Ph.D. degree in Mechanical Engineering from NUS in 2025. He is currently a Research Fellow in the Department of Mechanical Engineering at NUS. His research interests include intelligent transportation systems, multi-robot systems, and deep reinforcement learning.



Yue Zhou is an assistant professor in the Department of Engineering Science, Faculty of Innovation Engineering, at Macau University of Science and Technology. He obtained B.Eng., M.S., and Ph.D. from Tongji University, University of Nebraska, and Queensland University of Technology, respectively. He conducted post-doctoral research in New York University and Hong Kong Polytechnic University. His research interests mainly include Intelligent Transportation Systems, connected automated vehicles, traffic operations and control, traffic flow theory, transportation networks, and traffic safety.

Yiwei Wu is currently a Research Assistant Professor in the Department of Logistics and Maritime Studies at The Hong Kong Polytechnic University. She obtained the Ph.D. degree from the same department in 2024. She also holds an M.Phil. degree from the same department (2022), a Master's degree from Shanghai University (2020), and a Bachelor's degree from Shanghai Maritime University (2017). Yiwei's research interests include shipping operations management, port planning and operation, zero-carbon and green shipping, transportation and logistics optimization, and intelligent transportation.

Edward Chung received the bachelor's degree (Hons.) in civil engineering and the Ph.D. degree from Monash University. He was a Professor at the Queensland University of Technology (QUT) and the Director of the Smart Transport Research Centre, QUT. He is currently a Professor of intelligent transport systems (ITS) with the Department of Electrical Electronic Engineering, The Hong Kong Polytechnic University. With an extensive background as both an engineer and an accomplished academic researcher, he has garnered significant experience working on national and international projects. Notably, he held positions, such as the Senior Research Scientist at Australian Road Research Board, the Manager of Infrastructure Analysis and Modelling at the Victorian Department of Infrastructure, Australia, a Visiting Professor at the Centre for Collaborative Research, University of Tokyo, and the Head of the ITS Group, LAVOC, EPFL, Switzerland.

Guillaume Sartoretti joined the Mechanical Engineering department at the National University of Singapore as an Assistant Professor in 2019. Before that, he was a Postdoctoral Fellow in the Robotics Institute at Carnegie Mellon University. He received his Ph.D. degree in robotics from EPFL in 2016. He also holds a B.S. and an M.S. degree in Mathematics and Computer Science from the University of Geneva. He is interested in the emergence of collaboration/cooperation in large groups of intelligent agents making individual choices based on their local understanding of the world. Guillaume was a Manufacturing Futures Initiative (MFI) postdoctoral fellow at CMU in 2018-2019, was awarded an Amazon Research Awards in 2022, as well as an Outstanding Early Career Award from NUS' College of Design and Engineering in 2023.