machines

MDPI

Article

Learning Scheduling Method for Mixed Traffic at Autonomous Intersections Without Reliable Explicit Turn Information of Human-Driven Vehicles

Xuhao Yue [1], Feng Peng [2], Zejian Deng [3,*], Haoran Li [4], Chuan Sun [4,5], Hao Shi [5] and Haiming Sun [6]

[1] Shenzhen Lianming Power Co., Ltd., Shenzhen 518105, China
[2] The Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China
[3] Department of Data and Systems Engineering, The University of Hong Kong, Hong Kong SAR 999077, China
[4] Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215134, China
[5] School of Rail Transportation, Soochow University, Suzhou 215200, Chin
[6] Hubei Zhongcheng Industrial Technology Research Institute Co., Ltd., Shiyan 442000, China
* Correspondence: z49deng@hku.hk

Abstract

At autonomous intersections in mixed traffic, where Connected and Autonomous Vehicles (CAVs) coexist with Human-Driven Vehicles (HDVs), scheduling must remain effective even when a reliable explicit HDV turning intention is not available sufficiently early for the scheduling decision. This paper proposes a learning-based platoon scheduling method for this information-limited setting. CAVs and HDVs are organized into mixed platoons or an HDV group, and a state-augmentation function is designed to encode their temporal relationship while preserving the priority of uncontrollable HDVs. An enumeration (EN)-based expert demonstration mechanism is further integrated into the training process to provide high-quality experience. In the representative training run reported in the manuscript, the expert-assisted model achieved a peak average reward approximately 14% higher than the model trained without demonstrations. Across the tested demand cases, the learned scheduler also obtained travel-cost performance close to the EN benchmark. The scope of the method is explicitly limited to the modeled lane-keeping and sensing assumptions; robustness to random seeds, unexpected HDV maneuvers, communication delays, and additional safety metrics requires dedicated validation.

Keywords: deep reinforcement learning; autonomous intersection; mixed traffic; platoon scheduling

Academic Editor: Zheng Chen

Received: 29 August 2026
Revised: 20 September 2026
Accepted: 21 September 2026
Published: 26 September 2026

1. Introduction

Intersections represent pivotal nodes within the intricate framework of road traffic systems, where vehicle movement conflicts frequently precipitate accidents and congestion, rendering intersections critical bottlenecks and underscoring the imperative for sophisticated intersection management strategies. With the progressive evolution of autonomous driving technologies and Vehicle-to-Everything (V2X) communication systems, cutting-edge methodologies have been proposed to address the dual challenges of safety and operational efficiency at autonomous intersections. On one hand, the deployment of

Vehicle-to-Infrastructure (V2I) communication technologies allows roadside units (RSUs) to dynamically interact with CAVs, utilizing real-time vehicle data to execute intelligent scheduling decisions. Whether a vehicle operates in isolation or as part of a platoon, roadside units can assign optimal crossing times, thereby optimizing traffic flow. On the other hand, autonomous driving technology enables the precise trajectory planning of vehicles as they pass intersections, and the synchronization of this planning with scheduling further amplifies both the safety and throughput of intersection crossings. Nonetheless, the widespread integration of CAVs into existing transportation networks is anticipated to be an incremental process, leading to the coexistence of CAVs and HDVs in a mixed traffic environment for the foreseeable future. Consequently, investigating cooperative multi-vehicle control strategies in a mixed traffic environment, while accounting for the behavioral idiosyncrasies of HDVs, holds scientific value and practical significance.

The practical importance of this problem can also be quantified. The U.S. Federal Highway Administration reports that intersections account for roughly one-quarter of traffic fatalities and about one-half of traffic injuries each year [1]. A recent large-scale intersection study further notes that the United States has more than 320,000 signalized intersections and that drivers incur approximately $22.9 billion in annual direct and indirect congestion costs at these locations [2]. Meanwhile, mixed traffic is expected to persist rather than disappear rapidly. A market-penetration scenario reported by Oh et al. [3] used HDV/Level-3-AV/Level-4-CAV shares of 94/6/0 in 2025, 88/10/2 in 2030, and 84/13/3 in 2035, indicating that human-driven vehicles can remain dominant even while automated and connected vehicles are gradually introduced. These observations motivate a scheduling method that can operate over a wide range of CAV penetration rates instead of relying on a near-term transition to fully connected traffic.

At autonomous intersections, where vehicles are no longer governed by traditional traffic signals, a substantial challenge arises in terms of efficient scheduling. To address this complexity, researchers have proposed several methods, including the “first-come, first-served”(FCFS) method [4], the “first-in, first-out”(FIFO) method [5], the auction method [6], the longest queue first (LQF) method [7], and the batch processing method [8]. While these methods establish a predetermined order of passage for all vehicles, their inherent rigidity significantly limits their capacity to respond dynamically to fluctuating traffic conditions. Consequently, more sophisticated optimization-based scheduling methods have been developed to enhance intersection performance. These methods can generally be classified into three main categories: heuristic optimization methods [9–14], reservation-based optimization methods [15–20], and mathematical programming methods [12,21–28]. However, to further refine the effectiveness of these optimization models, it is important to note that these categories are not mutually exclusive and often integrate with one another. For instance, heuristic rules have been incorporated to relax constraints within mathematical programming frameworks [12], and reservation-based strategies have been employed to develop mixed-integer linear programming models [15,17]. In contrast to the aforementioned rule-based methods, these methods hold the potential to substantially enhance traffic efficiency by striving to achieve optimal scheduling through systematic, algorithmic search processes.

In heuristic optimization scheduling methods, scheduling models are constructed based on heuristic strategies that take into account not only the objective functions and constraints [9–11], but also the solution spaces and search strategies [10,12,13,29]. The objective function serves as the cornerstone of the optimization model, and its design presents one of the most significant challenges to be addressed. Wu et al. [9] tackled the optimization scheduling problem by formulating it as a variant of the Traveling Salesman Problem (TSP), where vehicles are analogous to cities and the headways between vehicles represent the distances between these cities. Li et al. [29] proposed a conflict graph tree

search method which can compress the solution space and reduce repeated calculations. Yao et al. [12] devised a heuristic rule aimed at reducing the binary constraints within the optimization scheduling model to improve computational efficiency. Xu et al. [13] initially developed a tree-structured solution space for the vehicle scheduling problem and introduced two heuristic rules designed to improve the efficiency of the Monte Carlo Tree Search (MCTS) method. The results demonstrated that, compared to the classical MCTS method, the enhanced version can identify the optimal solution within 0.1 s. Similar efforts have been documented in the literature [14]. Different from the above research, Yang et al. [30] designed a distributed scheduling model based on a long-horizon self-organization policy.

Unlike the heuristic optimization scheduling methods discussed previously, reservation-based optimization models must account for additional factors such as reservation protocols [15], vehicle turning maneuvers [16], traffic demand [17], and CAV penetration rates [20]. The reservation process must strictly follow a predefined protocol to ensure proper operation. Levin et al. [15] introduced an innovative reservation protocol that reduces computational complexity by autonomously determining when a vehicle should resend a reservation request and by evaluating the feasibility of such reservations in real time. Vehicle turning maneuvers are a crucial factor in reservation scheduling, as they significantly affect the allocation of limited conflict area. Minimizing the impact of turning movements remains a major challenge. To address this, Choi et al. [16] developed a reservation-based optimization model that incorporates three vehicle passage plans for a bidirectional three-lane intersection scenario. In contrast, Chen et al. [18] proposed a model that allows each entrance lane of the intersection to support all movement directions. While this configuration offers flexibility, it also increases the complexity of both the modeling and solving process, particularly for larger intersections. To address the scheduling challenges posed by high traffic demand, Yu et al. [17] introduced a batch processing strategy within the reservation optimization scheduling model. However, the aforementioned research did not consider the influence of HDVs on scheduling process. To bridge this gap, Zhong et al. [20] proposed a spatiotemporal reservation optimization model that incorporates the penetration rates of CAVs, addressing the mixed traffic scheduling problem more comprehensively.

In mathematical programming methods, scheduling models are typically formulated as Linear Programming (LP) or Mixed-Integer Linear Programming (MILP) models. For instance, Pei et al. [23] developed a MILP model for vehicle scheduling, and to enhance computational efficiency, they introduced a state transition method based on Markov Decision Processes (MDP). This method primarily operates by reducing the complexity of the scheduling solution space, allowing for the optimal scheduling solution to be derived through a state space search. Enhancing intersection capacity becomes significantly more feasible when vehicles are organized into platoons rather than passing the intersection individually [31]. To address this, several researchers have incorporated platoon formation into the scheduling process. Li et al. [19] integrated the FCFS method with a non-conflicting lane selection mechanism to schedule vehicle platoons. However, this study did not consider platoon formation during the scheduling process. In response to this limitation, Deng et al. [25] developed a MILP model that optimizes both the passage order and size of platoons, accounting for the dynamic formation process. Cong et al. [28] transformed the optimization scheduling problem into a packing optimization problem based on a virtual platoon strategy. Similar efforts have been documented in the literature [21,24,32–34], contributing further to the field.

All three methods face the challenge of improving model-solving efficiency. Scheduling optimization problems are classified as discrete optimization problems, which are inherently more complex to solve compared to continuous optimization problems [35].

While researchers have endeavored to mitigate this complexity by scaling up scheduling tasks, simplifying model constraints, and accelerating search processes, achieving significant improvements in model-solving efficiency remains a formidable challenge. Recently, there has been increasing interest in leveraging Deep Reinforcement Learning (DRL) methods to tackle vehicle scheduling problems. Lombard et al. [36] developed a Deep Q-Network (DQN) model specifically for vehicle scheduling at intersections. In their work, the state is represented as a 2D image encompassing all vehicles and road geometry, while the action space includes all possible passage orders. The reward function integrates two metrics: the average waiting time of all vehicles and the total number of vehicles that have successfully cleared the intersection. To further enhance the fairness of right-of-way allocation, Wu et al. [26] incorporated traffic fairness into their learning model, evaluating both traffic efficiency and fairness as reward objectives. In contrast to Lombard's method, Wu et al. represented the state as a binary matrix detailing the number of vehicles in each lane and their respective IDs. Similar research efforts are documented in reference [37].

In mixed traffic, the quality and availability of traffic information are themselves part of the control problem. Heterogeneous sensing and communication sources may contain missing or asynchronous data; for example, Min et al. [38] combined dedicated short-range communication data with vehicle-detection-system data to estimate traffic speed in missing-data segments. At the intersection-control level, Chen et al. [39] explicitly considered uncertain and delayed shared states and coupled robust set estimation with control to maintain safety under mixed traffic. These studies show that state estimation, information fusion, and coordination should not be treated as independent modules. Accordingly, the present work adopts a deliberately conservative information setting: the RSU can observe HDV position and speed, but the scheduler does not rely on a reliable explicit turn-intention label being available sufficiently early for the current scheduling decision.

Recent studies have also continued to develop mixed-traffic intersection coordination. Jiang et al. [40] formulated platoon formation and scheduling for mixed traffic at unsignalized intersections, while recent DRL studies have incorporated explicit right-of-way coordination or heterogeneous conflict graphs for mixed traffic [41,42]. These developments provide important contemporary context for evaluating the proposed state representation and expert-guided learning strategy.

DRL methods have effectively mitigated the limitations inherent in traditional optimization scheduling methods by substantially reducing both the modeling and computational complexity, thus garnering substantial interest from researchers. Nevertheless, several limitations persist in the current research:

- Zhou et al. [43] proposed a reasoning-graph reinforcement-learning method for mixed-traffic scheduling and assumed that HDV turning intentions are predetermined. In practice, HDV intention is not necessarily completely unobservable: it may be inferred probabilistically from turn signals, lane position, trajectory history, or perception-based prediction. However, these cues can be unavailable, delayed, or uncertain at the time when an upstream scheduling decision must be made. Therefore, the problem considered in this paper is more precisely defined as scheduling without relying on a reliable explicit HDV turn-intention label at the decision time. This conservative setting complements intention-prediction approaches rather than claiming that HDV intention can never be estimated.
- The previously discussed learning scheduling methods demonstrate limited success in substantially improving model learning efficiency. Specifically, the reward gradients considerably hinder the learning process, particularly during the later stages of model training. At this stage, although the model has learned most of the optimal actions, a small number of optimal actions remain unlearned. As training progresses, the reward gradients between optimal and suboptimal actions diminish significantly,

deteriorating the model's discriminative capacity. Consequently, the model requires a prohibitively large number of iterations to capture these marginal reward differentials, often leading to convergence on local optima.

To address the identified research gaps, an innovative learning scheduling method is designed. The primary advantages of this method are elucidated through the following two aspects:

- A state-augmentation function and a learning scheduling method are developed for the above information-limited setting. In contrast to a scheduler that requires a deterministic HDV turn label, the proposed coordination layer optimizes the passage order of mixed platoons without using reliable explicit HDV turn intention as an input. It can also determine the relative passage order of multiple mixed platoons within one scheduling decision.
- An expert demonstration mechanism is devised and integrated into the model training process, which significantly augments the model's learning performance. In the previously discussed research [26,36,37,43], the learning models are required to explore the action space independently and painstakingly, devoid of guidance and demonstrations from the expert model.

2. Problem Statement and Analysis

This paper develops a learning model for mixed traffic at autonomous intersections and a training method intended to improve its scheduling performance. Figure 1 summarizes the technical roadmap. Existing learning schedulers may use deterministic HDV turning information [43]. Here, the operational condition is different: the RSU is assumed to obtain the position and speed of an HDV from roadside sensing, but a reliable explicit turn-intention label is not assumed to be available sufficiently early for the current scheduling decision. Probabilistic intention estimates from turn signals, lane position, or trajectory history could be incorporated in future extensions, but they are intentionally not required by the present scheduling layer. To remain conservative under this information limitation, vehicles are coordinated at the platoon level and, for scheduling units with potentially conflicting movements, only one unit is assigned the conflict-area right of way at a time. Platoon scheduling is therefore decomposed into platoon formation and passage coordination, while the CAV penetration rate affects both operations because only CAV motion can be directly commanded by the RSU.

Building upon existing research, a comprehensive learning model must be constructed considering three fundamental components: state, action, and reward. In existing research, researchers typically use the vehicle's speed, position, and turn information as the state for the learning model [26,36,37,43]. However, this operation may not be suitable for the scheduling problem addressed in this paper. Directly concatenating the raw kinematic data of all vehicles into the RL state vector would inevitably lead to the curse of dimensionality. Such an excessively high-dimensional state space hinders the agent's ability to extract underlying traffic patterns, severely degrading learning efficiency. Therefore, designing a robust state augmentation function is imperative to encode the topological and temporal features of platoons concisely. On the other hand, the learning model should leverage HDV information to facilitate optimal coordination of CAV platoons.

The platoon scheduling task falls under the category of discrete action problems.

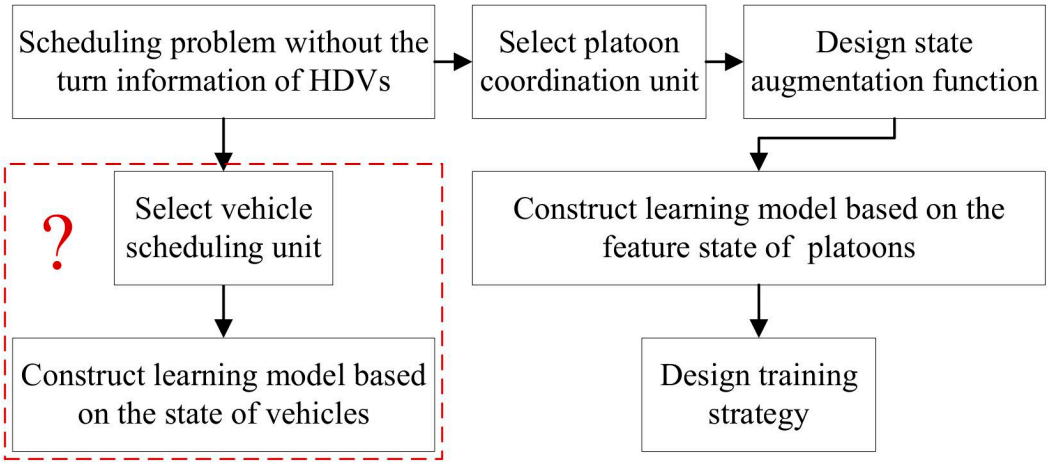


Figure 1. Technology roadmap analysis.

In a discrete action space, each action is represented in a discrete form with no inherent relationships among them. Enhancing decision-making performance relies on the learning model's ability to identify optimal actions across all observed states. Such optimal experiential data are crucial for updating neural networks, enabling the learning model to master optimal strategy. To address this challenge, two primary operations can be considered: (1) enhancing the exploration capabilities of the learning model to identify optimal actions within the action space, and (2) providing high-quality experiential data to facilitate model learning. To supply the model with high-quality experiential data, it is necessary to solve the problem and achieve global optimal solutions. This requirement entails that the problem must be represented as an explicit function, as implicit functions pose significant challenges to direct resolution.

To further elucidate the aforementioned problems, an autonomous intersection scenario is introduced, as illustrated in Figure 2. Each approach comprises two entry lanes: the inner lane accommodates left-turning and through traffic, while the outer lane serves right-turning and through traffic. Both CAVs and HDVs are considered within this scenario, with the CAV penetration rate ranging from 0% to 100%. When CAVs enter the communication area, the RSU facilitates platoon formation, passage coordination, and longitudinal speed planning for these vehicles, which subsequently execute the issued commands. In contrast, due to the lack of communication capabilities, the driving behavior of HDVs remains uncontrolled. CAVs in different lanes, provided there is no collision risk in the conflict area, may be assigned to the same platoon. However, longitudinal formation mode is not accounted for. The CAVs enclosed within the blue dashed box represent a platoon. Upon entering the conflict area, the platoon will disband. Since a platoon may consist of CAVs traveling in various directions, only one platoon is permitted to pass the conflict area at a time.

The safety scope of this abstraction depends on explicit operational assumptions. HDVs are assumed to remain in their assigned entrance lane while approaching and traversing the modeled conflict area; roadside sensing provides their current position and speed; and an unexpected HDV occupancy of the conflict area triggers the CAV yielding fallback described later in the paper. The proposed scheduler therefore does not claim a formal collision-avoidance guarantee for arbitrary unmodeled maneuvers, such as an abrupt lane change, route change, or a sensing outage. These cases require an additional intention-estimation/reachability layer, such as the robust estimation-to-control framework in Chen et al. [39].

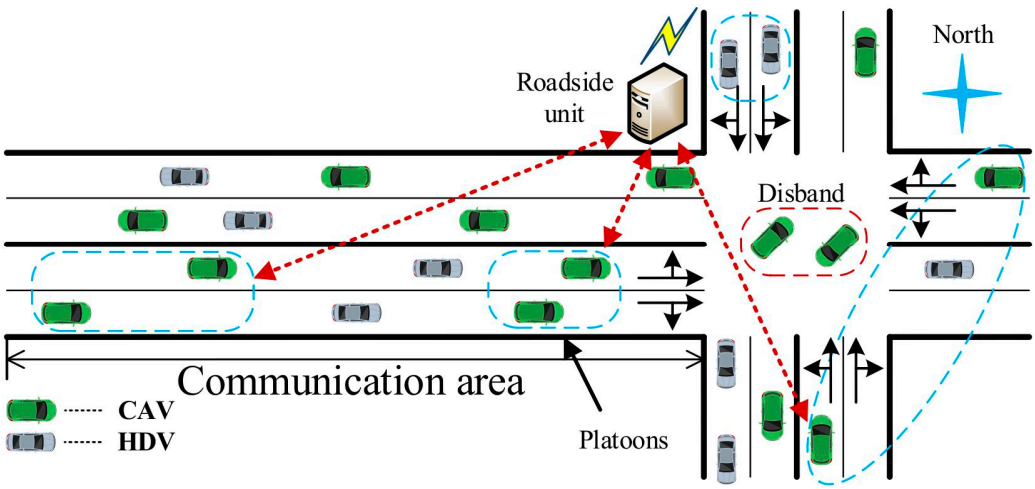


Figure 2. Autonomous intersection.

3. Learning Scheduling Method

3.1. Dueling Deep Q-Network

Deep reinforcement learning operates within the framework of a MDP, where an agent continuously interacts with its environment to learn an optimal policy aimed at maximizing cumulative rewards. The agent observes the current state of the environment and, based on the objective of reward maximization, selects the most advantageous action. Upon executing this action, the environment transitions to a new state and delivers feedback to the agent in the form of this updated state and the corresponding reward signal, enabling the agent to iteratively improve its decision-making capability. For instance, in a reinforcement learning problem, the state observed by the agent at time t is $s_t \in S$. Based on its current decision-making capability, the agent executes an optimal action $a_t \in A$ and receives a reward $r_{t+1} = r(s_t, a_t)$, leading the environment to transition from state s_t to a new state s_{t+1}. For a given state s_t, the agent's decision is governed by the probability distribution of actions, which is defined by the policy $\pi(a_t \mid s_t)$. When the agent performs action a_t at time t, the probability of the environment transitioning from state s_t to s_{t+1} is represented by $p(s_{t+1} \mid s_t, a_t)$. The agent aims to maximize the future cumulative discounted reward expressed as $r_t^\gamma = \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k})$, where γ is the discount factor, constrained by $0 \le \gamma \le 1$, and $\gamma^k r$ denotes the discounted reward at time $t+k$. The discount factor translates future rewards into their present value, indicating the importance of future rewards to the current state. The agent derives the optimal action and maximizes cumulative rewards by approximating both the state value function and the action value function. The state value indicates the expected cumulative reward starting from state s_t and following policy π, and it can be expressed as:

$$V_\pi(s_t) = \mathbb{E}_\pi\left[r_t^\gamma \mid s_t\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) \middle| s_t\right] \quad (1)$$

The action value, in turn, represents the expected cumulative reward obtained by executing action a_t in the current state s_t while following policy π, which is expressed as:

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi\left[r_t^\gamma \mid s_t, a_t\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) \middle| s_t, a_t\right] \quad (2)$$

In the Q-learning method, the agent evaluates all possible actions a_t based on the current state s_t and selects the optimal action that maximizes cumulative rewards.

This selection is governed by the following iterative update equation:

$$Q_\pi(s_t, a_t) = Q_\pi(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q_\pi(s_{t+1}, a) - Q_\pi(s_t, a_t)\right) \quad (3)$$

where α represents the learning rate. The Q-learning method relies on a Q-table to store the Q-values associated with each action. While this method is effective for relatively simple problems, it becomes inefficient when dealing with large-scale or continuous state-action spaces due to the significant memory requirements for storing and updating the Q-table, as well as the extensive time needed to explore the state space comprehensively. To address these challenges, Mnih et al. [44] proposed the use of deep neural networks (DNNs) to approximate Q-values, eliminating the need for an explicit Q-table. In the DQN method, the Q-values are approximated by a neural network, and its parameters are updated by minimizing the following loss function:

$$L(\theta^Q) \approx \left(r_{t+1} + \gamma \max_a Q_\pi(s_{t+1}, a \mid \theta^Q) - Q_\pi(s_t, a_t \mid \theta^Q)\right)^2 \quad (4)$$

This method diverges from the traditional Q-learning method, where a Q-table is explicitly used to store Q-values for each action. DQN introduces experience replay to address the instability associated with neural network updates. During the training process, a fixed-size experience buffer is established to store the agent's interaction data with the environment, represented as $(s_{t+1}, a_t, r_{t+1}, s_t)$. As training progresses, the buffer fills with data, and once the buffer reaches capacity, older data is replaced. The stored experience data are then used to update the neural network parameters, with the experience replay mechanism ensuring that past interactions are fully leveraged to improve learning. Both DQN and Q-learning methods employ the ϵ-greedy strategy to balance exploration and exploitation in action selection. While this method accelerates the learning model's convergence toward the optimal policy, excessive use can result in overestimation of Q-values.

In contrast to traditional Q-learning and DQN methods, the Dueling Deep Q-Network (2DQN) introduces a state value function, which separates the evaluation of state importance from action-specific values. In many scenarios, the inherent value of a state is more critical than the action selected within that state. By incorporating a state value function, the 2DQN method more effectively captures the significance of states without relying solely on action-specific value estimates. This architecture proves particularly advantageous in situations where the differences between actions in a given state are minimal. In the 2DQN method, the Q-network is divided into two components: the state value function and the advantage function. The state value function is solely dependent on the state, whereas the advantage function is contingent upon both the state and the action. The advantage function quantifies the relative benefit of executing action a in the current state s, as compared to other possible actions. To prevent the overestimation of actions, the advantage function is often normalized by subtracting its mean value across all actions in the state. Therefore, the combined value function can be expressed as:

$$Q(s, a) = V(s, w, \alpha) + \left(A(s, a, w, \beta) - \frac{1}{|A|}\sum_{a'} A(s, a', w, \beta)\right) \quad (5)$$

where $V(\cdot)$ denotes the state value function, w represents the shared parameters of the Q-network, α refers to the parameters of the state value function, $A(\cdot)$ signifies the advantage function, and β represents the parameters associated with the advantage function.

3.2. Decision-Making Framework

To address the platoon scheduling problem at an autonomous intersection, a learning scheduling method is designed, which incorporates both platoon formation and passage coordination. The proposed method employs a centralized decision-making framework consisting of two distinct layers, as illustrated in Figure 3. The first layer handles platoon

formation, while the second layer manages platoon passage coordination. According to this structure, CAVs are initially organized into platoons, followed by the optimization of their passage order through coordination. Both the formation and coordination operations are executed by the proposed method, which is implemented within the RSU.

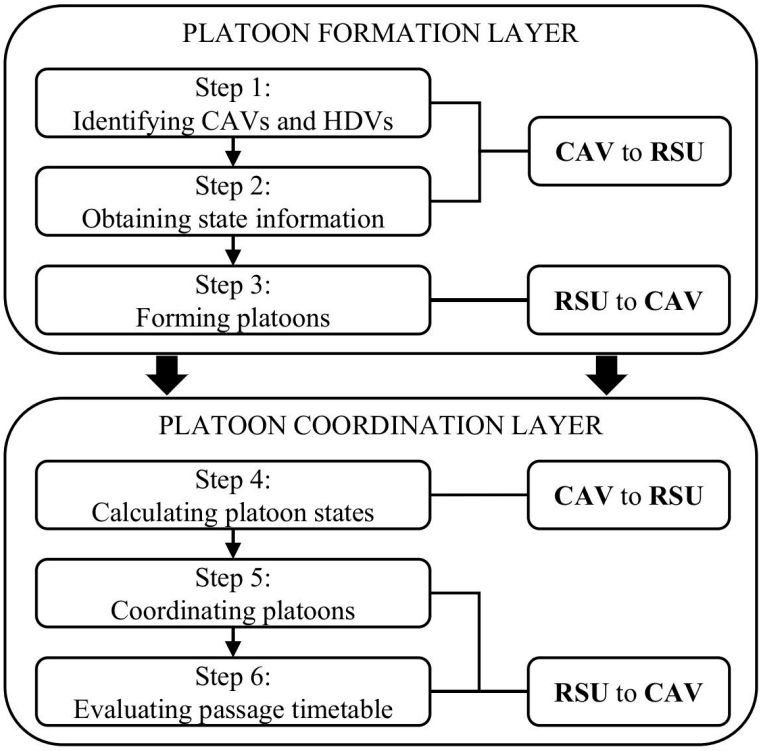


Figure 3. Centralized decision-making framework.

In the first layer, the optimal platoon formation problem is tackled through three main steps (i.e., Step 1 to Step 3). In Step 1, when CAVs and HDVs enter the communication area of the autonomous intersection, the RSU identifies all CAVs by receiving their passage requests, while utilizing sensor devices to detect the presence of HDVs. In Step 2, the RSU collects the current position, speed, and turning information of the CAVs, along with the position and speed of the HDVs. Finally, in Step 3, the RSU forms the CAVs and HDVs into mixed platoons or HDV groups based on the state information received from CAVs and the sensor devices, subsequently communicating the results to each CAV. Consequently, three types of platoons exist in the autonomous intersection scenario: CAV platoon, HDV group, and mixed platoon consisting of both CAVs and HDVs. To simplify the discussion in the subsequent sections of this paper, from this point forward, both CAV platoon and mixed platoon will be collectively referred to as mixed platoon.

In the second layer, the optimal platoon coordination problem is addressed through four key steps (i.e., Step 4 to Step 6). In Step 4, the RSU calculates the current state of each platoon based on the transmitted data. In Step 5, the RSU determines the optimal passage order for each platoon, taking into account their respective driving states. In Step 6, the RSU evaluates passage time for each mixed platoon or HDV group, based on the established order, and communicates this information to the CAV members of each mixed platoon.

3.3. Methodology

Before coordinating the platoons, it is essential to first form platoons for all vehicles within the communication area. The process is outlined in Algorithm 1:

Algorithm 1 Platoon formation algorithm
Input The state of all vehicles V and lane set L = {1, 2, 3, …}

Output Optimal platoons P = {HDV-g₁, P₂, P₃, …, Pₙ}
Find lane l* ∈ L with the maximum number of vehicles
Divide lane l* into segment set S_{l*} = {s^1_{l*}, s^2_{l*}, …}, where each segment length is determined by the distance between consecutive vehicles' rear bumpers
for each unlabeled lane l ∈ L \ {l*} do
  Segment lane l into S_l based on the number and corresponding lengths of segments in S_{l*}
end for
for each ordinal index k do
  Initialize segment group G_k ← ⋃_{l∈L} s_l^k
  Group all CAVs and HDVs within G_k into vehicle group V_{G_k}
end for
for each vehicle group V_{G_k} do
  for each vehicle in V_{G_k} do
    Compare its arrival time with that of labeled vehicles in other groups
    if an adjacent group yields a smaller arrival time difference then
      Reassign the vehicle to that adjacent group
    end if
  end for
  Form CAVs into mixed platoons P_i based on minimal arrival time differences, ensuring non-conflicting turning movements
  Initialize remaining HDVs into a separate group HDV-g₁
  for each HDV in V_{G_k} do
    if the preceding vehicle is a CAV then
      Append the HDV to the preceding CAV's mixed platoon P_i
    end if
  end for
end for
Return P

Figure 4 illustrates a simple example focusing on the 4 lanes of an autonomous intersection. Since lane 8 contains the most vehicles, it is labeled and segmented accordingly. The lane is divided into four unequal-length segments based on vehicle positions, and Lane-1, Lane-2, Lane-7, and Lane-8 are segmented in a similar manner. These segments are grouped into four distinct groups. The group-1 includes Lane-1-1, Lane-2-1, Lane-7-1 and Lane-8-1. The group-2 includes Lane-1-2, Lane-2-2, Lane-7-2 and Lane-8-2. The group-3 includes Lane-1-3, Lane-2-3, Lane-7-3 and Lane-8-3. The group-4 includes Lane-1-4, Lane-2-4, Lane-7-4 and Lane-8-4. In group-1, there is a CAV. In group-2, there are two CAVs and an HDV. In group-3, there are two CAVs and an HDV. In group-4, there is a CAV and an HDV. In Step-5 and Step-6, the arrival times are evaluated based on the current positions and speeds of the CAVs. Figure 5 illustrates a case for platoon formation. In Figure 5, group-1 consists of two CAVs and two HDVs, and group-2 comprises two CAVs and two HDVs, which are divided into three parts: HDV group-1, mixed platoon-2 and mixed platoon-3. HDV-6 and HDV-8 are formed into mixed platoon-2 as the following vehicles behind CAV-7 and CAV-5, respectively. CAV-1 and CAV-4 are formed into the mixed platoon-3.

Once all vehicles have been formed into mixed platoons or an HDV group, the coordination problem of them must be addressed. Within the platoon coordination layer, the learning model assumes a pivotal role in Step-4 and Step-5. As illustrated in Figure 5, there are two mixed platoons and an HDV group. The driving behaviors of mixed platoons can be regulated. However, the behaviors of the HDV group remain outside the scope of control. Furthermore, coordination operations of the two mixed platoons and the HDV group do not require turning information but rely on position and speed.

Passage coordination belongs to a discrete decision-making problem. Consequently, the learning model is tasked with generating discrete actions to ascertain the optimal passage order. Thus, the action can be articulated as:

$$A = \left[a_1^{\text{HDV-g}_1}, a_2^{p_2}, a_3^{p_3}, \ldots, a_n^{p_n}\right], A \in \mathcal{A} \quad (6)$$

where A represents the action vector, where a denotes the specific ordinal in which a platoon passes the conflict area, and n signifies the total number of mixed platoons and HDV groups. The action space, $\mathcal{A}$, encompasses all possible passage orders for the platoons, with the total number of such orders being $n!$. Each element within an action vector is distinct, which effectively prevents potential collisions between platoons while also minimizing the complexity of the action space. For example, if there are five platoons required to pass the conflict area, a randomly selected passage order could be expressed as $A = [1,3,4,5,2]$. In this case, the passage order for HDV group-1 is 1, for mixed platoon-2

is 3, for mixed platoon-3 is 4, for mixed platoon-4 is 5, and for mixed platoon-5 is 2. The number of actions within $\mathcal{A}$ is 5!.

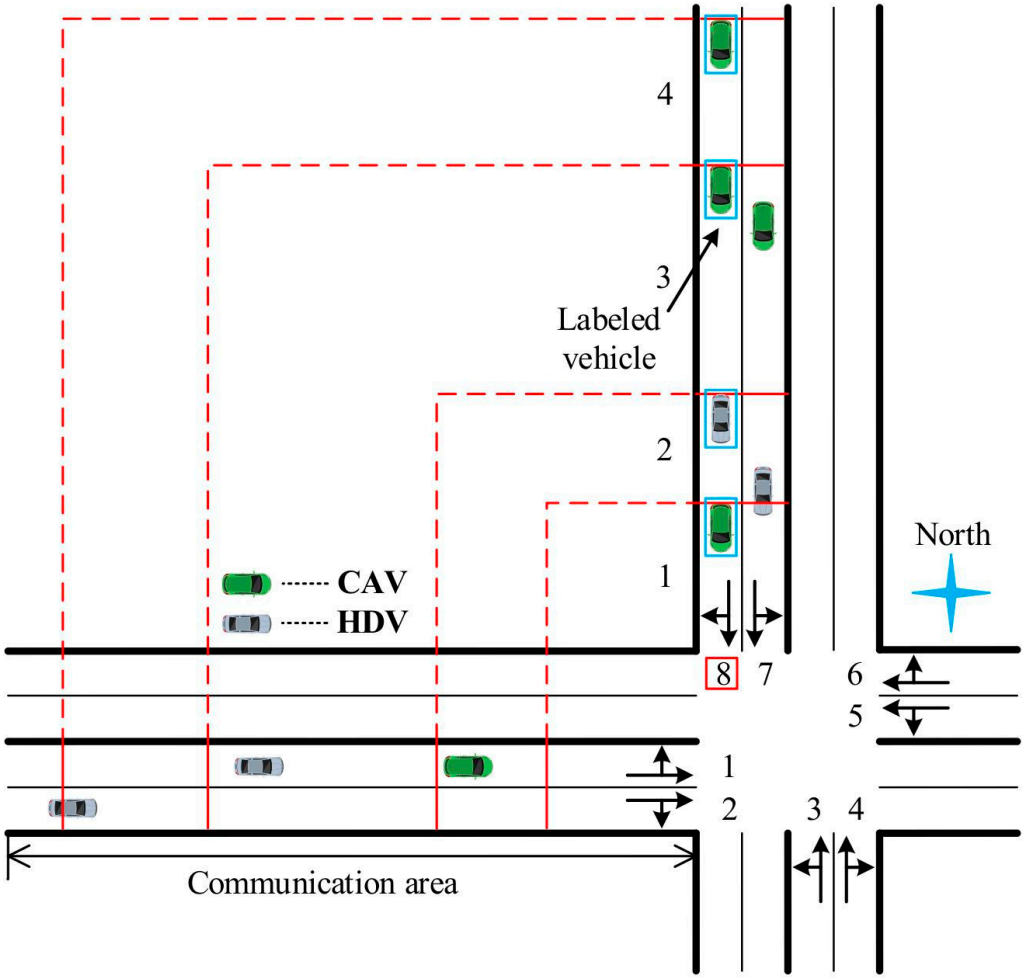


Figure 4. The case for segment grouping and vehicle grouping.

Setting the observation state is a crucial step in constructing the learning model, as the model can make optimal decisions only when it accurately understands the current platoon state. In the platoon coordination problem, the state information of the platoons is often redundant. For instance, the state information of a single vehicle includes its speed, position, and the time required to pass through an intersection. If there are five platoons, each with four vehicles requiring coordination, the state dimension needs to be set to 60 to accommodate the observation data. Such a large amount of data might hinder the model's ability to comprehend the state of targets, thereby affecting its learning efficiency. Selecting observation data that accurately represents the current driving state of the platoon is crucial. Additionally, it is essential to reflect the differences between platoons' states, as these differences help the model identify optimal actions.

During the coordination process, the passing timestamps of all mixed platoons can be flexibly adjusted. However, due to the lack of control over the HDV group, the time window for their passage remains immutable. Consequently, any mixed platoon with overlapping time windows with the HDV group is required to yield. The learning model must be capable of internalizing this fixed priority rule. More critically, the learning model must be able to make optimal decisions by processing the state data from both platoons. As illustrated in Figure 5, when HDV group-1 is given priority over mixed platoon-2, the subsequent passage order between mixed platoon-2 and mixed platoon-3 must be

optimized. Additionally, the model must be proficient in adapting its decision-making process to account for varying CAV penetration rates.

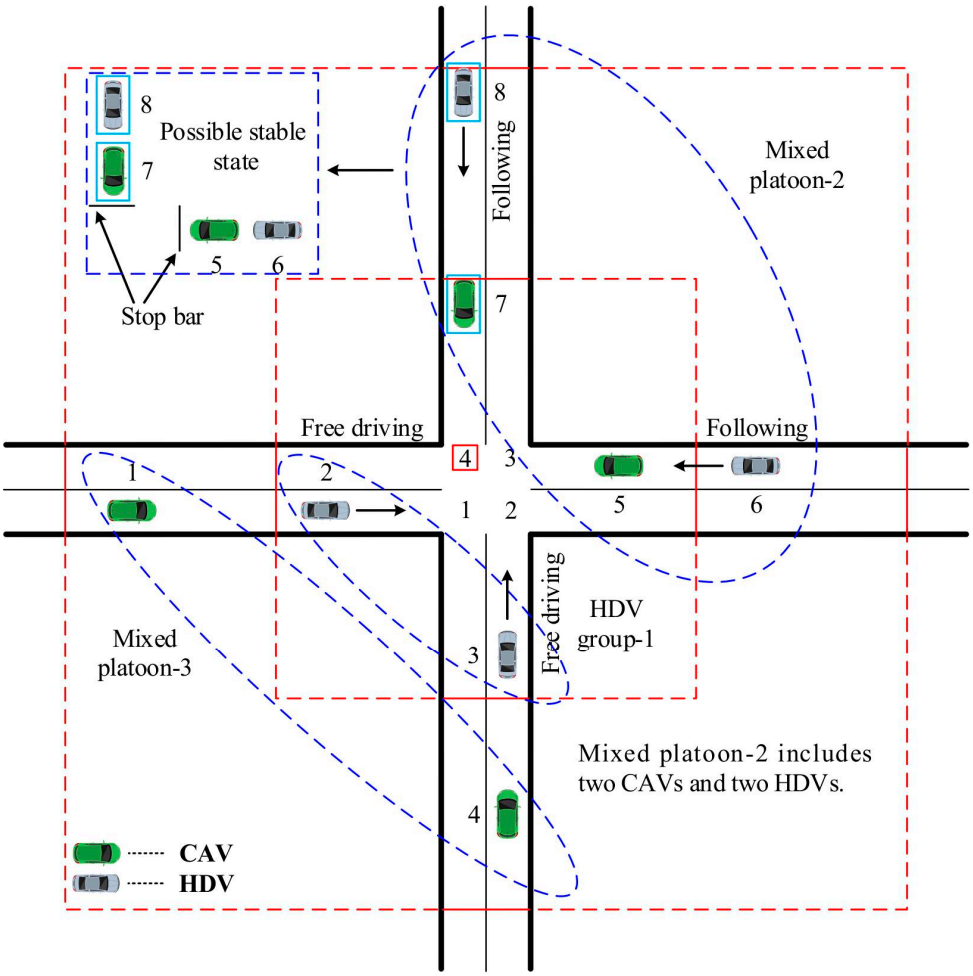


Figure 5. The case for forming platoons.

Based on the above analysis, an innovative state function is designed, denoted as:

$$F_s\left(\left[t_1^p, t_2^p\right]\right) = \begin{cases} t_2^p + T_m, & \text{if } \left[t_1^p, t_2^p\right] \cap \left[t_1^{\text{HDV-g}}, t_2^{\text{HDV-g}}\right]_1 \neq \varnothing, \\ t_2^p + T_m, & \text{if } t_1^p > t_2^{\text{HDV-g}} \text{ and } \exists \left[t_1^p, t_2^p\right]_{i \in n} \cap \left[t_1^{\text{HDV-g}}, t_2^{\text{HDV-g}}\right]_1 \neq \varnothing, \\ t_2^p, & \text{if } T_a < t_1^{\text{HDV-g}} \text{ and } f_o\left(t_2^p\right) < t_1^{\text{HDV-g}}, \quad t_2^p \in \{t_2^p \mid t_2^p < t_1^{\text{HDV-g}}\}. \end{cases} \tag{7}$$

$$\left[t_1^p, t_2^p\right] \in \left\{\left[t_1^{\text{HDV-g}}, t_2^{\text{HDV-g}}\right]_1, \left[t_1^p, t_2^p\right]_2, \left[t_1^p, t_2^p\right]_3, \ldots, \left[t_1^p, t_2^p\right]_n\right\}$$

where t_1^p and t_2^p are the evaluated arrival and departure timestamps of each mixed platoon, and $t_1^{\text{HDV-g}}$ and $t_2^{\text{HDV-g}}$ are the corresponding timestamps of the HDV group. Let $I_h = \left[t_1^{\text{HDV-g}}, t_2^{\text{HDV-g}}\right]$ and let

$$C = \left\{ i \mid \left[t_{1,i}^P, t_{2,i}^P\right] \cap I_h \neq \emptyset \right\}, \; T_m = \max\left(0, \; t_2^{\text{HDV-g}} - \min_{i \in C} t_{1,i}^P\right) \tag{8}$$

When$C \neq \emptyset$; otherwise $T_m = 0$. Thus, T_m is not an additional physical waiting time imposed on a vehicle. It is a state-feature translation determined by the earliest arrival timestamp among the platoons that overlap the HDV interval and by the HDV-group departure timestamp. For every directly overlapping platoon $i \in C, t_{2,i}^p + T_m \geq t_2^{\text{HDV-g}}$, which makes the HDV-priority relation explicit in the augmented state. Applying the same additive offset to the second category preserves the pairwise temporal order among

the affected platoons because, for any i and j, $t_{2,i}^{p} < t_{2,j}^{p}$ if and only if $t_{2,i}^{p} + T_m < t_{2,j}^{p} + T_m$. This is the reason that the same Tm is used rather than a platoon-specific offset.

The three conditions in Equation (7) are operational categories for platoons whose ordering relative to the HDV group must be encoded; they are not intended as an exhaustive partition of all possible time intervals. Category 1 contains a platoon whose nominal interval directly overlaps I_h. Category 2 contains a downstream platoon with $t_1^p > t_2^{\mathrm{HDV-g}}$ when at least one platoon in the same coordination episode belongs to Category 1; the common translation preserves the relative structure of this coupled scheduling set. Category 3 contains a platoon nominally before the HDV group that can still be completed before $t_1^{\mathrm{HDV-g}}$ after accounting for the currently available passage time T_a and the trial ordering function f_o. Platoons that are temporally separated from the HDV group and do not participate in these coupled cases retain their original departure timestamp as the state feature.

To elucidate the aforementioned state function, several cases are illustrated in Figure 6, where the red arrowed line signifies the passage interval of the HDV group, while the black arrowed line denotes the passage interval of the mixed platoon. In Figure 6a, the passage intervals of platoon-1 and platoon-3 overlap with that of HDV group-2, which possesses the right of way, necessitating that the other platoons yield. This precedence is attributed to the non-cooperative behavior of the HDV group-2 toward mixed platoons. Conversely, in Figure 6b, the passage interval of HDV group-1 remains entirely distinct from those of any other platoons, thereby granting HDV group-1 the right of way.

According to the third category, the passage intervals of certain platoons do not overlap with those of the HDV platoon. The condition $T_a < t_1^{\text{HDV-g}}$ indicates that the arrival timestamp of the HDV platoon is later than the available passage timestamp T_a, thereby establishing a temporal window during which other platoons may advance. However, these platoons must fulfill the condition $t_2^p \in \left\{ t_2^p \mid t_2^p < t_1^{\text{HDV-g}} \right\}$. The function $f_o(t_2^p)$ is capable of determining which platoon may pass the conflict area without overlapping with the passage interval of the HDV group. Furthermore, f_o can initialize a random passage order for these platoons and compute the departure timestamp for each under this randomized arrangement. If $f_o(t_2^p) < t_1^{\text{HDV-g}}$, the respective platoon is granted the right of way over the HDV group. In Figure 6c, HDV group-3 arrives later than the available passage timestamp T_a, and the passage intervals of the other two platoons do not overlap with it. According to the random order generated by $f_o(t_2^p)$, the passage interval of platoon-2 overlaps with that of HDV group-3, thus granting platoon-1 the right of way over HDV group-3.

Figure 6d illustrates why the nominal condition $t_2^p < t_1^{\mathrm{HDV-g}}$ alone is insufficient. If the current available passage timestamp T_a shifts a platoon's executable interval, that interval can still conflict with the HDV group. The function f_o therefore checks whether a candidate pre-HDV order can actually be completed before $t_1^{\mathrm{HDV-g}}$. When an overlap is present, T_m is computed by Equation (8). Its role is solely to separate the encoded state classes and expose the fixed HDV-priority relation to the Q-network; actual collision avoidance is enforced by the passage-order feasibility rules and the yielding mechanism, not by interpreting T_m as a physical safety headway.

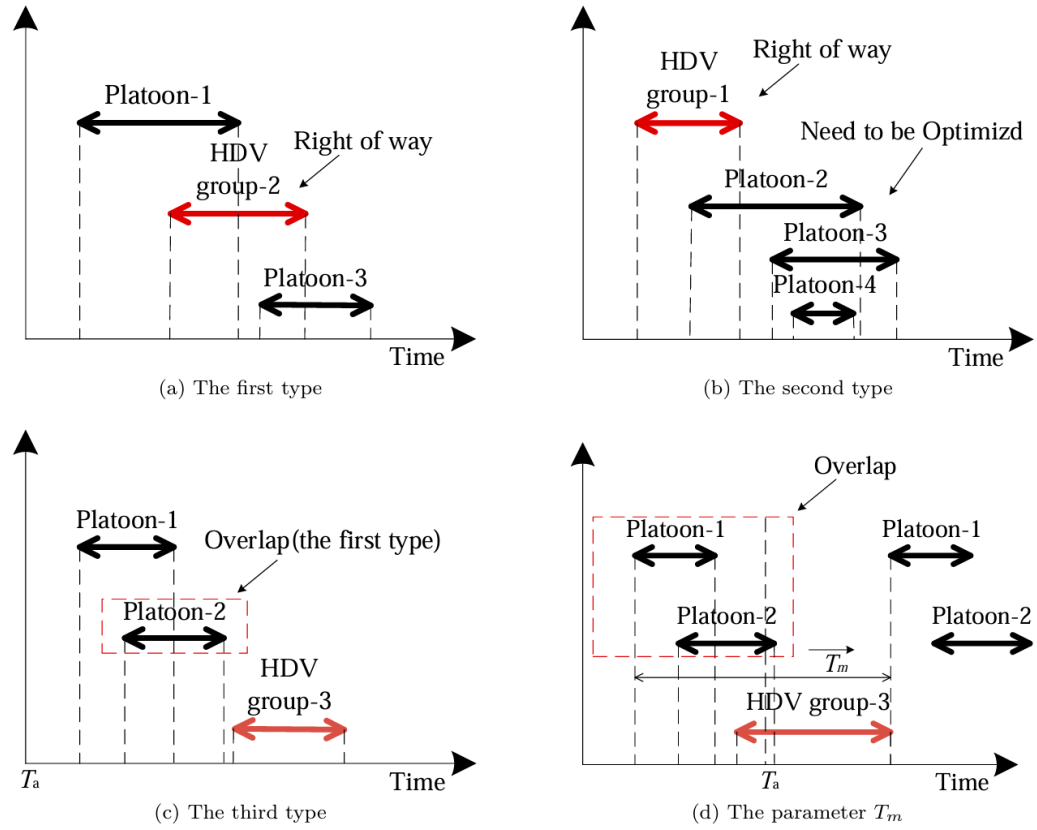


Figure 6. The illustration of state function.

The observation state can be set as:

$$O = \left[t_2^{\text{HDV-g}_1}, t_2^{p_2}, t_2^{p_3}, \ldots, t_2^{p_n}\right] \tag{9}$$

Based on Equations (7) and (9), the observation state vector conveys the following features to the learning model:

- If $t_2^{\text{HDV-g}} < t_2^{p_i}$, the HDV group is granted right of way over platoon-i, as their respective passage intervals do not overlap, and the departure timestamp of the HDV group precedes the arrival timestamp of platoon-i in the state augmentation process.
- Conversely, if $t_2^{\text{HDV-g}} > t_2^{p_i}$, platoon-$i$ is a candidate to pass before the HDV group. Because an apparently feasible nominal interval can become infeasible after accounting for T_a and the preceding platoon order, $f_o(\cdot)$ is used as a conservative temporal-feasibility check. It rejects a candidate pre-HDV ordering whenever the evaluated departure timestamp reaches or exceeds $t_1^{\text{HDV-g}}$. Therefore, f_o should be interpreted as reducing temporal overlap risk in the scheduling representation rather than, by itself, constituting a complete microscopic collision-avoidance guarantee. Figure 7a–d illustrate this filtering effect.

Building upon the aforementioned two features, mixed platoons can be classified into two distinct categories. The first category corresponds to cases where $t_2^{\text{HDV-g}} < t_2^{p_i}$, while the second pertains to cases where $t_2^{\text{HDV-g}} > t_2^{p_i}$. The passage order for these platoons must be optimized by the learning model, with the passage interval for the HDV group remaining invariant. Equations (7) and (9) also apply to 100% CAV penetration rate. Under 100% or high CAV penetration rate, there may be no HDV group, so $t_2^{\text{HDV-g}} = 0$. The state augmentation process is illustrated in Algorithm 2.

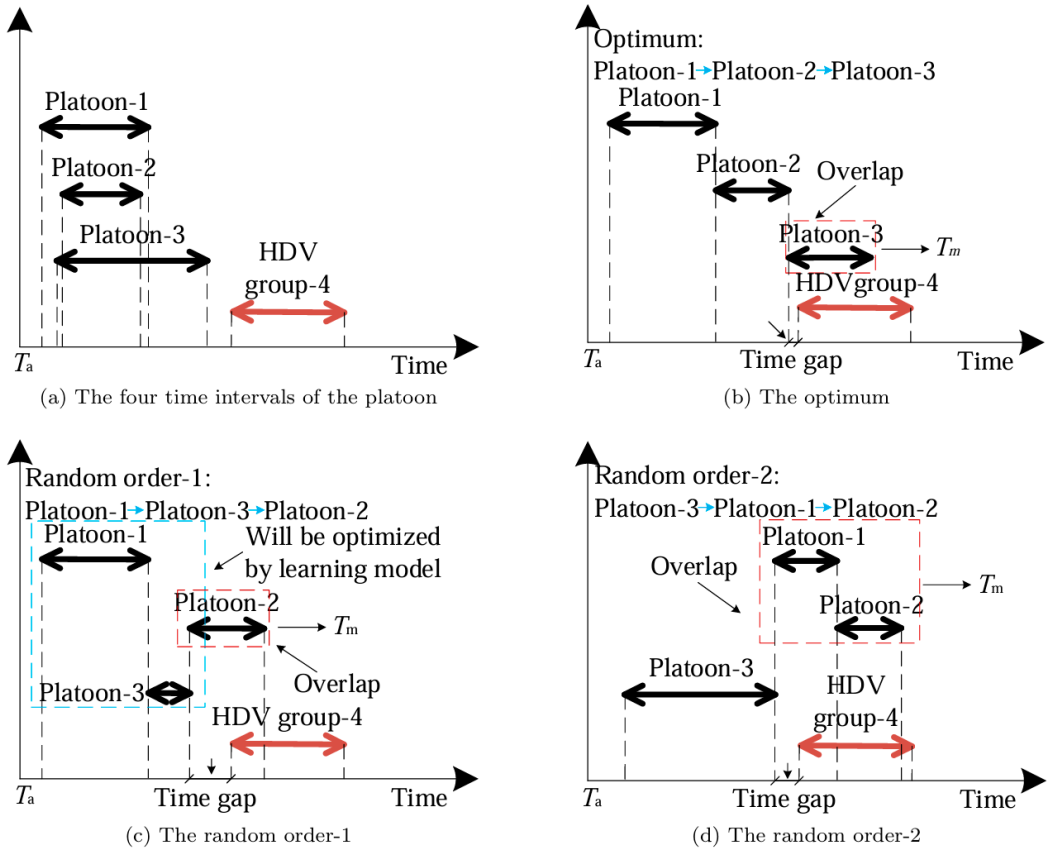


Figure 7. The illustration of $f_o(t_2)$.

Algorithm 2 State augmentation algorithm

Input The arrival and departure timestamps of all mixed platoons and HDV group.
Output [t₂^{HDV-g₁}, t₂^{P₂}, t₂^{P₃}, …, t₂^{Pₙ}]
if There is no HDV group then
  Set t₂^{HDV-g₁} ← 0
else
  for j = 2 to n do
    if [t₁^{P_j}, t₂^{P_j}] ∩ [t₁^{HDV-g}, t₂^{HDV-g}]₁ ≠ ∅ then
      Set t₂^{P_j} ← t₂^{P_j} + T_m
    end if
    if t₁^{P_j} > t₂^{HDV-g} and ∃ [t₁^{P_i}, t₂^{P_i}]_{i∈n} ∩ [t₁^{HDV-g}, t₂^{HDV-g}]₁ ≠ ∅ then
      Set t₂^{P_i} ← t₂^{P_i} + T_m
    end if
  end for
end if

How to set the reward to ensure that the learning model can understand the above features and then make an optimal decision that yields the maximum reward is important. Granting the right of way between mixed platoons and the HDV group without hesitation, therefore, the reward function can be denoted as:

$$r(O,A)=\begin{cases}\alpha, & \text{if } \left(t_2^{p_i}<t_2^{\text{HDV-g}} \text{ and } a_i^{p_i}>a_1^{\text{HDV-g}}\right) \text{ or } \\ & \left(t_2^{p_i}>t_2^{\text{HDV-g}} \text{ and } a_i^{p_i}<a_1^{\text{HDV-g}}\right),\\ -\sum\left(t_p^d-t_1\right), & else\end{cases} \tag{10}$$

where $\alpha<0$ is the penalty assigned to an action that violates the encoded HDV-priority relation, and t_p^d is the evaluated departure timestamp under the candidate passage order. Let

$$J_{\max}=\sup_{A\in\mathcal{A}_{\text{feas}}}\sum_p\left(t_p^d-t_{1,p}\right) \tag{11}$$

be an upper bound on the total passage-time cost among feasible actions in one scheduling decision. A sufficient reward-separation condition is $\alpha<-J_{\max}$; under this condition, every priority-violating action receives a smaller one-step reward than every

feasible action. This inequality explains the required magnitude of the penalty, although it is not, by itself, a proof of global convergence of the nonlinear function approximator.

The priority rule is additionally treated as a hard feasibility rule: when the first branch of Equation (10) is triggered, the candidate action is rejected and is not executed in the traffic environment. The second branch therefore optimizes efficiency only within the admissible scheduling set. Safety is not intentionally traded against travel time through a weighted reward term; longitudinal collision avoidance and conflict-area fallback rules are imposed separately by the car-following/control model and the yielding logic described in Section 4. This separation between a hard safety layer and an efficiency reward is consistent with recent constraint-aware reinforcement-learning vehicle-control studies [42]. The distinction between the state variable $t_2^{p_i}$ and the reward variable t_p^d is also important: the former is computed before choosing a passage order, whereas the latter is evaluated after applying the candidate order and the trajectory-planning model.

Traffic demand at intersections is subject to real-time fluctuations, and as such, learning models must be capable of addressing platoon scheduling challenges of varying scales. However, in deep reinforcement learning methods, once the dimensions of state and action spaces are defined, they are fixed and cannot be adapted. For instance, if the observation space is set to a three-dimensional state, the model is only capable of solving coordination problems for three platoons and is unable to extend its coordination capabilities to platoons with different sizes. To mitigate this limitation, a virtual encoding method is proposed. To mitigate the dimensional rigidity inherent in DRL architectures, we propose a virtual encoding strategy. This approach bifurcates the state and action vectors into real and virtual components. Real components encode the dynamic states and executable actions of physical platoons, whereas virtual components act as scalable placeholders, accommodating variable traffic demands without necessitating structural alterations to the neural network. Similarly, in the action vector, only the real component is executable, while the virtual component is not. Despite this distinction, both the real and virtual components of the state and action vectors are utilized in the neural network's update process, enabling the model to dynamically scale and address platoon coordination problems of varying sizes. Equation (12) presents the virtual encoding:

$$\begin{cases} O = \left[\underbrace{o_1, o_2, o_3}_{\text{actual}}, \underbrace{o_4, o_5}_{\text{virtual}}\right] \\ A = \left[\underbrace{a_1, a_2, a_3}_{\text{actual}}, \underbrace{a_4, a_5}_{\text{virtual}}\right] \end{cases}, \tag{12}$$

where o_1, o_2, o_3 and a_1, a_2, a_3 denote the states and actions of the actual scheduling units, while o_4, o_5 and a_4, a_5 denote fixed-dimension virtual entries. A virtual platoon contains one virtual CAV, and its travel-time feature is evaluated with the same trajectory-evaluation procedure used for a single-CAV platoon; the associated virtual action is discarded before any command is applied to the physical traffic environment.

The Markov-property implication can be stated explicitly. Let the fixed-dimensional encoding be $\tilde{O} = g(O, N)$, where O is the physical Markov state, N is the current number of actual scheduling units, and g is a deterministic padding/encoding function that depends only on the current decision state. Let $P_N(\tilde{A})$ remove the virtual entries from a network action before execution. Then the encoded transition satisfies

$$\Pr\left(\tilde{O}', \mid \tilde{O}\tilde{A}\right) = \Pr\left(g, (O', N') \mid OP_N(\tilde{A})\right) \tag{13}$$

which introduces no dependence on pre-O history provided that the virtual-state assignment is deterministic in the current physical state. Thus, fixed-dimensional encoding does not inherently destroy the Markov property. It can nevertheless affect function approximation and sample efficiency; therefore, the effect of the virtual-state design should

be examined by a dedicated state-representation ablation rather than assumed to be bias-free.

During the initial phase of training a 2DQN model, the model has not yet mastered the optimal policy, resulting in the output of actions that are largely stochastic, yielding relatively low rewards. Gradually, the experience pool becomes increasingly enriched with high-quality actions and corresponding substantial rewards. These tuples are then sampled to update the model's neural network, facilitating further policy optimization. Over time, the model converges towards the optimal policy, and the rewards stabilize within a specific range. Two primary factors can impede the model's ability to explore and identify the optimal order. First, in a platoon coordination problem, there exists only one optimal order that minimizes passage time. Given the large action space, it becomes apparent that the learning model must conduct an extensive search to locate the optimal order. Suboptimal orders may not yield sufficiently high reward values for the learning model, and only a select few suboptimal orders may provide relatively higher rewards. Second, the distinction between suboptimal order and optimal order may be subtle. On one hand, altering the passage order between any two platoons could lead to large passage times. On the other hand, certain suboptimal orders may differ considerably from the optimal one, yet still result in passage times that are nearly indistinguishable.

An expert demonstration model, based on the EN method, was developed and integrated into the training process of the 2DQN model, as depicted in Figure 8. The improved learning model is called the 2DQN-Expert model. During the training process, the expert model and the 2DQN model operate in parallel, engaging in simultaneous decision-making. The expert model retrieves all relevant state information of the platoons from the training environment and performs an exhaustive enumeration of all possible passage orders to determine the optimal one. The reward for the optimal passage combination is computed in accordance with Equation (10), and the corresponding experience tuple is subsequently added to the experience pool. Notably, although the expert model acquires state information from the training environment, it cannot execute the optimal passage order. Conversely, the 2DQN model extracts state information from the training environment, and the actions it generates are executed within the training environment. The state, rewards, and actions are also stored in the experience pool as experience tuples. Consequently, the experience pool contains two categories of data: the first consists of the expert model's demonstration data, while the second comprises the training experience data of the 2DQN model. Compared to the 2DQN model, the expert model demonstrates stable performance, reliably producing optimal orders and securing high rewards.

The expert mechanism injects globally optimal permutation examples generated by the EN solver into the replay data for this scheduling problem. Exploration and demonstration influence the learner through different channels: the 2DQN agent continues to select the action executed in the environment according to its exploration policy, whereas the expert action is not executed and affects learning only when its stored tuple is sampled for a gradient update. In this way, expert experience supplements rather than replaces online exploration.

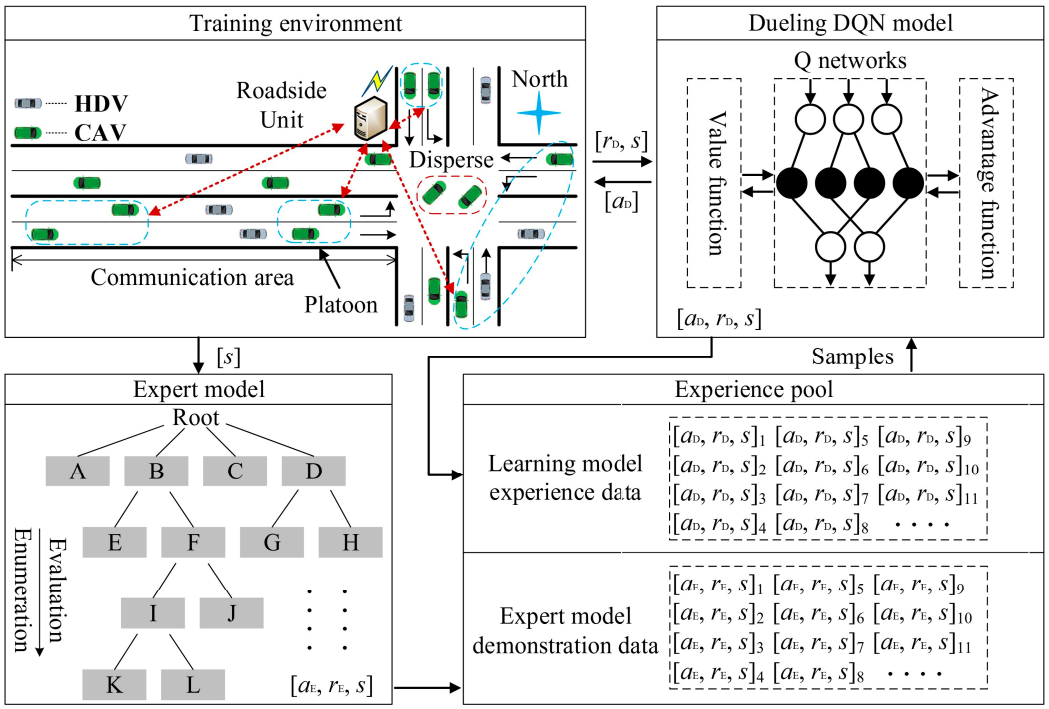


Figure 8. Training process with expert demonstration mechanism.

4. Longitudinal Trajectory Planning

After determining the order of platoon passage, the RSU must formulate longitudinal trajectory plans for the mixed platoons approaching the conflict area from the entrance lanes. For each platoon, the vehicle positioned closest to the conflict area is designated as the leading vehicle (LV). The longitudinal trajectory optimization model is then constructed based on the dynamic state of this LV.

$$\begin{aligned} &\min_{u_i(t)} \frac{1}{2}\int_{t_i^1}^{t_i^2} u_i(t)^2\, dt \\ &\text{subject to: } u_{\min} \le u_i(t) \le u_{\max} \\ &\qquad v_{\min} \le v_i(t) \le v_{\max} \\ &\qquad p_i(t_i^1) = p_i^1, \quad v_i(t_i^1) = v_i^1 \\ &\qquad p_i(t_i^2) = p_i^2, \quad v_i(t_i^2) = v_i^2 \\ &\qquad t \in [t_i^1, t_i^2] \end{aligned} \tag{14}$$

where i is the index, $u(t)$ represents the acceleration, $u_{\min}$ and $u_{\max}$ denote the lower and upper bounds of the acceleration, respectively, and v indicates the speed. Additionally, p^1 corresponds to the initial position of the LV, v^1 to the initial velocity of the LV, p^2 to the final position of the LV, v^2 to the final velocity of the LV, t^1 to the initial timestamp for the longitudinal trajectory planning, t^2 to the final timestamp, and t represents the planning horizon.

The optimization goal of Equation (14) is to minimize the cumulative acceleration within the specified planning horizon. This objective not only seeks to enhance passenger comfort but also aims to reduce fuel consumption caused by frequent acceleration and deceleration. The Hamiltonian method [45] is employed to solve Equation (14), while a consensus control method [46] is utilized to conduct the platoon control task. To guarantee the passage safety of all vehicles through the intersection, the following clarifications are provided:

- Upon entering the conflict area, a mixed platoon will disband, and CAVs will pass the conflict area at a constant speed. Control tasks for CAVs will continue to be conducted.

- Regardless of whether an HDV is part of a mixed platoon or an HDV group, its longitudinal trajectory on the entrance lane and within the conflict area will be governed by the car-following model [47].
- Once an HDV enters the conflict area, the right of way for HDVs with conflicting movement will be assigned using the FCFS method.
- Under a low CAV penetration rate, evaluating the passage time of the HDV group based on the optimal passage order may introduce error. Such error could lead to both HDVs and CAVs with conflicting trajectories entering the conflict area simultaneously. In these instances, CAVs must yield to HDVs and are permitted to proceed only once HDVs have departed the conflict area.

Safety scope of the platoon-level abstraction**.** Let $J_k = \left[\tau_k^{\text{in}}, \tau_k^{\text{out}}\right]$ denote the actual conflict-area occupancy interval of scheduling unit k, and let $c_{ij} = 1$ indicate that two distinct units have geometrically conflicting movements. The scheduling layer enforces the sufficient inter-unit condition

$$c_{ij} = 1 \implies J_i \cap J_j = \emptyset \tag{15}$$

except for HDVs already treated inside the same HDV group, whose interaction is handled by the microscopic car-following/FCFS rules. If an HDV is observed in the conflict area outside its predicted interval, conflicting CAVs do not execute their nominal crossing action and instead yield until the HDV has cleared the conflict area.

Proposition 1. *Under the modeled assumptions that (i) vehicles remain in their as-signed lane through the conflict area, (ii) the microscopic longitudinal model/controller prevents rear-end collisions within a scheduling unit, (iii) Equation (15) is satisfied for distinct conflicting scheduling units, and (iv) the CAV yielding fallback is triggered when an HDV occupies the conflict area unexpectedly in time, the platoon-level scheduling abstraction prevents modeled inter-unit conflict-area collisions.*

Proof. A collision between two distinct units with conflicting movements requires simultaneous occupancy of their common conflict region. Condition (iii) excludes that simultaneous occupancy under the nominal schedule, while condition (iv) blocks a conflicting CAV crossing when the observed HDV occupancy deviates from the nominal interval. Longitudinal interactions within a unit are delegated to condition (ii). Hence, the modeled inter-unit collision modes are excluded under assumptions (i)–(iv). □

This proposition deliberately states the scope rather than claiming safety under arbitrary HDV behavior. Abrupt lane changes, route changes not represented by the current lane geometry, perception outages, and non-negligible sensing/communication delays fall outside the present guarantee. These cases therefore remain outside the empirical scope of the current experiments and are identified as a limitation and future extension, for which an uncertainty-aware intention/reachability layer and dedicated safety stress testing would be appropriate [39].

5. Experimental Evaluation

To facilitate the training and evaluation processes, the SUMO simulation platform is employed, given its capability to comprehensively assess the efficiency of various traffic management methods. This section presents four experiments aimed at rigorously testing the designed learning model. In the first experiment, the learning model, both with and without the expert demonstration mechanism, will undergo training, with a comparative analysis of their results conducted to underscore the beneficial impact of the expert demonstration mechanism in enhancing decision-making efficacy. The second experiment

involves the training of alternative models, followed by a comprehensive comparison and critical analysis of all performance metrics. In the third experiment, the proposed learning model will be evaluated under 100% CAV penetration, with its scheduling performance benchmarked against established methods to emphasize its superior efficiency. Lastly, in the fourth experiment, the performance of the learning model will be examined under varying CAV penetration rates, thereby showcasing its robustness and adaptability in addressing the mixed traffic scheduling problem.

5.1. Training Environment

To train the learning model, a simulation environment is first established, with the environment parameters detailed in Table 1. Each episode of the simulation runs for 500 s. The communication range is set to 250 m. Scheduling operations are exclusively applied to the first two vehicles in each lane that enter the communication range, while vehicles already scheduled within this range are excluded from additional operations. The range of free-flow speed is set at 10–11 m/s, and the maximum speed limit on the road is 11 m/s. The acceleration is constrained within the range of −6 m/s², 3 m/s². Traffic demand varies between 50 veh/l/h and 400 veh/l/h, where the unit veh/l/h denotes vehicles per lane per hour.

Table 1. Simulation environment parameters.

Parameter	Value
Communication range	250 m
Simulation time	500 s
Minimum acceleration	−6 m/s²
Maximum acceleration	3 m/s²
Range of free flow speed	10–11 m/s
Maximum speed	11 m/s
Minimum traffic demand	50 veh/l/h
Maximum traffic demand	400 veh/l/h
Range of CAV penetration rate	0–100%

The hyperparameters for the 2DQN model are presented in Table 2. The discount factor is set to 0.99, and the learning rate is 0.001. The initial and final epsilon values are 1.0 and 0.01, respectively. The replay pool has a capacity of 200,000 tuples, and the mini-batch size is 256. The state and action dimensions are both 7. The Q-network uses three shared fully connected layers with 256 neurons per layer and then separates into value and advantage streams, each containing a 256-neuron fully connected layer. Training is conducted for at most 1000 episodes.

Table 2. Hyperparameters of 2DQN model.

Parameter	Value
Discounted factor	0.99
Learning rate	0.001
Initial epsilon	1.0
Final epsilon	0.01
Experience pool size	200,000
Batch sample size	256
Q Network layer	5
Shared neural network layer	3
Value function Network layer	1
Advantage function Network layer	1

Neuron	256
State dimension	7
Action dimension	7
Maximum episode count	1000

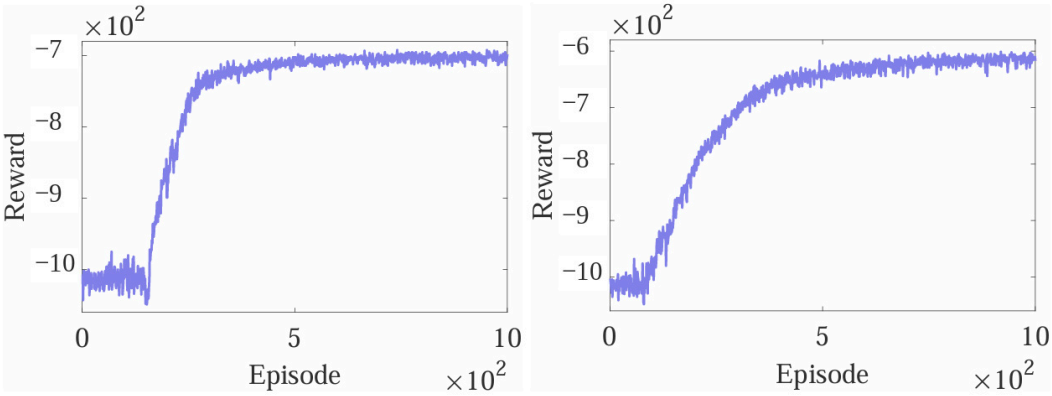
Evaluation scope and robustness. The figures and tables reported in this study are based on the simulation runs described above. Because independent repeated-seed results are not available for every learning configuration, the present analysis does not claim statistical significance. Instead, robustness is examined descriptively across the operating conditions already included in the study, including multiple traffic-demand levels, multiple CAV penetration rates, comparisons among DQN, DDQN, PPO, and SAC implementations, and comparison with the EN benchmark. These scenario sweeps evaluate sensitivity to operating conditions, but they are not treated as substitutes for independent-seed uncertainty estimates. Accordingly, the reported 14% reward increase is described only as an observation from the representative recorded training run.

The experiments also distinguish the effect of expert guidance by comparing 2DQN with 2DQN-Expert and comparing several alternative DRL architectures. However, an additional raw-state ablation that isolates the state-augmentation function alone is not included. We therefore avoid attributing the full performance gain to any single component and limit the corresponding conclusion to the combined framework evaluated here. Recent mixed-traffic right-of-way and heterogeneous-graph DRL studies are cited as relevant references [41,48].

For safety, the study evaluates efficiency only within the modeled lane-keeping, sensing, longitudinal-control, temporal-separation, and CAV yielding assumptions. The efficiency metrics are not presented as empirical evidence of robustness to unexpected HDV maneuvers. Likewise, the SUMO experiments do not include an explicit V2I communication-delay model or hardware-specific inference benchmark. We therefore describe the proposed policy as avoiding online exhaustive enumeration after training, rather than claiming experimentally validated real-time deployment on a specific RSU platform.

5.2. Training Performance Comparison

Figure 9 compares one recorded training run of the 2DQN and 2DQN-Expert models. In this run, the 2DQN reward became stable shortly after approximately 500 episodes, whereas the 2DQN-Expert curve stabilized after approximately 600 episodes. The 2DQN-Expert model reached a peak average reward of about −600 compared with about -700 for 2DQN, corresponding to the approximately 14% relative improvement reported in the original manuscript. This result is reported as a within-run observation rather than as a statistical superiority claim. Its interpretation is supported descriptively by the additional demand-level, penetration-rate, algorithm, and EN-benchmark comparisons reported in the following subsections. The slower stabilization of the expert-assisted curve may be related to the additional diversity introduced into the replay data; because the current records do not include a demonstration-ratio sensitivity experiment, we do not assign a unique causal mechanism to this difference.

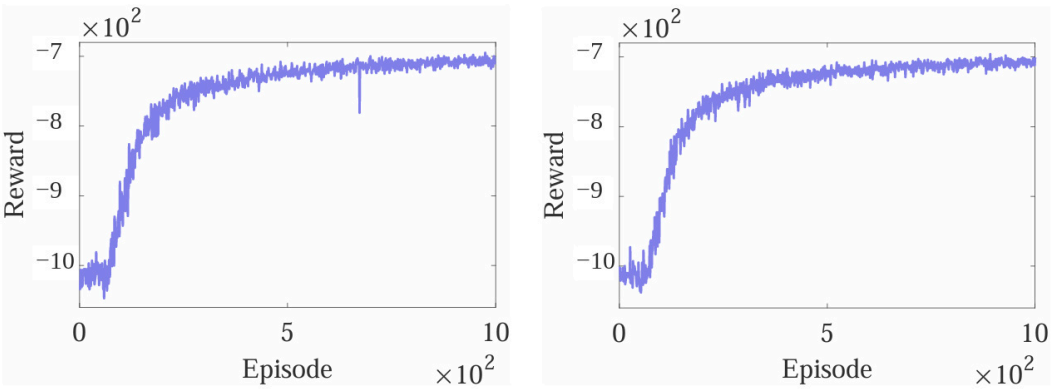


(a) The historical reward of 2DQN model (b) The historical reward of 2DQN-Expert model

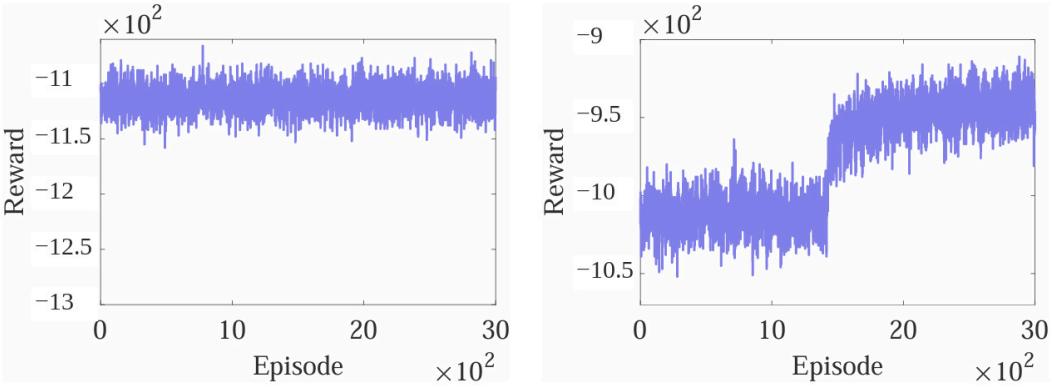
Figure 9. Training performance comparison between the 2DQN model and the 2DQN-Expert model.

To further assess the training efficacy of the 2DQN-Expert model, the DQN model, Double Q-Network (DDQN) model [49], Proximal Policy Optimization (PPO) model [50], and Soft Actor-Critic (SAC) model [51] are constructed and trained within the same simulation environment. To comprehensively demonstrate the training performance of the PPO and SAC models, the maximum number of episodes was set to 3000. Figure 10 illustrates the historical reward for each model. The PPO model exhibited the lowest reward values, fluctuating within a range of −1150 to −1080, and it failed to achieve convergence throughout the entire training process. In contrast, the SAC model demonstrated superior performance in terms of both reward magnitude and convergence rate when compared to the PPO model. After approximately 1500 episodes, the SAC model's reward values began to converge gradually, ultimately stabilizing within the range of −980 to −920. It is evident that the DQN and DDQN models outperformed both the PPO and SAC models in terms of training efficacy. The training results of the DQN and DDQN models were notably similar, with both models commencing convergence after approximately 100 episodes and achieving stable rewards after 500 episodes. However, the DQN model displayed a degree of instability around 680 episodes. While the convergence speed of both the DQN and DDQN models is slower than that of the 2DQN model, they surpass the 2DQN-Expert model. Nevertheless, the reward values achieved by the DQN and DDQN models remain lower than those obtained by the 2DQN-Expert model.

These algorithmic baselines compare the underlying learning architectures under the same local implementation, but they do not by themselves isolate the contribution of the proposed state design or establish superiority over recent mixed-traffic intersection frameworks. The additional ablation and recent-method comparisons specified above are therefore necessary for a complete attribution of performance gains.



(a) The historical reward of the DQN model (b) The historical reward of the DDQN model



(c) The historical reward of the PPO model (d) The historical reward of the SAC model

Figure 10. Training performance of the DQN model, the DDQN model, PPO model and SAC model.

5.3. Decision-Making Performance Comparison

Upon the completion of the 2DQN-Expert model's training, it is essential to rigorously assess its decision-making efficacy, specifically its learning performance. The Adaptive Signal Control (ASC) method, along with the FCFS, LQF, and EN methods, will be evaluated under varying traffic demand with a full CAV penetration rate. The decision-making effectiveness of these methods will be systematically compared to that of the 2DQN-Expert model. Ma and Yu et al. [52,53] employed \$6/h to quantify vehicle delay costs. Inspired by their research, the present section adopts \$6/h and \$1.5/L as the unit travel costs for vehicle delay and fuel consumption, respectively, with total travel cost serving as the primary evaluation metric. Table 3 illustrates the performance outcomes of all methods with full CAV penetration rate, while Tables 4–7 present a detailed comparative result.

Table 3. Test results of all methods under different traffic demand with full CAV penetration rate.

Demand (veh/h/lane)	50	100	200	300	400
ASC	2.6	4.2	15.0	38.1	42.7
	18.9	20.1	24.7	34.4	36.9
FCFS	3.6	4.0	5.2	6.1	11.1
	14.2	14.3	15.3	16.1	20.2
LQF	6.0	6.1	6.5	7.3	12.4
	16.5	15.7	16.2	17.4	21.6
EN	2.7	2.6	2.6	3.4	9.6
	13.9	13.9	14.0	14.9	20.7
2DQN-Expert	2.4	2.9	2.7	3.8	8.8
	13.8	13.9	13.8	15.0	20.0

Table 4. Comparison results of ASC method and 2DQN-Expert method.

Demand (veh/h/lane)	Delay (%)	Fuel (%)	Cost (%)	ASC ($/h/lane)	2DQN-Expert ($/h/lane)
50	−7.7%	−27.0%	−25.0%	3.3	2.5
100	−31.0%	−31.0%	−31.0%	3.7	2.6
200	−82.0%	−44.1%	−59.0%	6.1	2.5
300	−90.0%	−56.4%	−74.6%	11.3	2.9
400	−79.4%	−45.8%	−64.4%	12.4	4.4

Table 5. Comparison results of FCFS method and 2DQN-Expert method.

Demand (veh/h/lane)	Delay (%)	Fuel (%)	Cost (%)	FCFS ($/h/lane)
50	−33.3%	−2.8%	−9.3%	2.7
100	−27.5%	−2.8%	−8.5%	2.8
200	−48.1%	−9.8%	−20.0%	3.1
300	−37.7%	−6.8%	−15.7%	3.4
400	−20.7%	−1.0%	−8.3%	4.8

Table 6. Comparison results of LQF method and 2DQN-Expert method.

Demand (veh/h/lane)	Delay (%)	Fuel (%)	Cost (%)	FCFS ($/h/lane)
50	−60.0%	−16.4%	−28.6%	3.4
100	−52.5%	−11.5%	−23.5%	3.3
200	−58.5%	−14.8%	−27.9%	3.5
300	−48.0%	−13.8%	−24.4%	3.8
400	−29.0%	−7.4%	−15.6%	5.2

Table 7. Comparison results of EN method and 2DQN-Expert method.

Demand (veh/h/lane)	Delay (%)	Fuel (%)	Cost (%)	FCFS ($/h/lane)
50	−11.1%	−0.7%	−2.5%	2.5
100	+11.5%	+0.0%	+1.9%	2.5
200	+3.9%	−1.4%	−0.6%	2.5
300	+11.8%	+0.7%	+2.8%	2.8
400	−8.3%	−3.4%	−5.0%	4.6

The 2DQN-Expert method demonstrates superior performance compared to the ASC method across different levels of traffic demand for two primary reasons. First, the ASC method lacks the capability to coordinate vehicles in platoons, which leads to increased headways between vehicles and substantial time losses. Second, the ASC method is unable to integrate longitudinal trajectory optimization, making it challenging to achieve fuel efficiency. Furthermore, vehicles often need to stop before the stop line while waiting for clearance, which extends their dwell time in the conflict area, resulting in inefficient throughput and heightened fuel consumption. Both the FCFS and LQF methods are based on fixed rules, which fail to adapt to dynamic traffic conditions, thus limiting their ability to make optimal real-time decisions. Consequently, their performance is inferior to that of the 2DQN-Expert method. Table 7 provides a comparison between the EN method and the 2DQN-Expert method, with negligible differences observed. At traffic demand levels of 100–300 veh/h/lane, the 2DQN-Expert method incurs a 2% increase in costs. However, at demand levels of 50 veh/h/lane and 400 veh/h/lane, the costs are reduced by 2.5% and

5%, respectively, when compared to the EN method. Based on these results, it is evident that the 2DQN-Expert method's decision-making capability approaches that of the EN method. Given that the EN method yields globally optimal solutions, this indicates the effectiveness of the 2DQN-Expert learning process.

5.4. The Impact of CAV Penetration Rate on 2DQN-Expert Model

Twelve mixed-traffic cases were evaluated using three demand levels and four CAV penetration rates. In Table 8, the percentage columns are relative changes with respect to the corresponding 100% CAV case at the same demand level. Specifically,

$$\Delta D(p) = \frac{D(p) - D(100\%)}{D(100\%)} \times 100\%, \qquad \Delta F(p) = \frac{F(p) - F(100\%)}{F(100\%)} \times 100\% \tag{16}$$

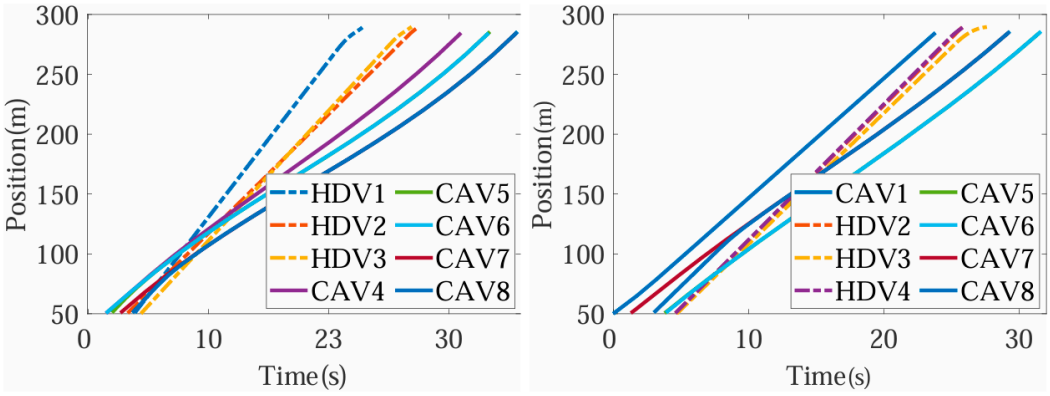
where D and F denote average delay and fuel consumption, respectively, and p is the CAV penetration rate. Therefore, the column originally labeled "Delay (%)" should be read as a relative delay increase rather than an absolute delay percentage.

The original results show a clear increase in fuel consumption as penetration decreases, whereas delay is not strictly monotonic at the two lower demand levels. For example, at 100 veh/h/lane, the recorded delay changes from 5.8 s at 80% penetration to 5.4 s at 20%. This non-monotonic value should not be interpreted as evidence that reducing CAV penetration improves delay. Under light demand, a larger HDV group resolved by the FCFS microscopic rule can produce shorter realized headways in a particular stochastic run, while the loss of trajectory optimization still increases fuel use. Because the current table contains recorded single-run values, the small non-monotonic differences at light demand may also reflect traffic-generation randomness. We therefore use Table 8 as a descriptive sensitivity analysis across penetration-rate scenarios and do not infer a statistically significant monotonic delay relationship from these individual values.

Figure 11 illustrates two right-of-way allocations at 100 veh/h/L and 60% CAV penetration. In Figure 11a, HDV-1, HDV-2, and HDV-3 pass before the CAV platoons; in Figure 11b, CAV-1 passes first because its evaluated passage interval is temporally separated from the HDV-group interval.

Table 8. Test results of the 2DQN-Expert model under different traffic demands and CAV penetration rates.

Demand (veh/h/lane)	Penetration (%)	Delay (s)	Relative Delay (%)	Fuel (ml)	Relative Fuel (%)
100	80	5.8	+100.0%	17.8	+28.1%
	60	5.8	+100.0%	18.5	+33.1%
	40	5.7	+96.6%	19.2	+38.1%
	20	5.4	+86.2%	19.9	+43.2%
200	80	6.4	+137.0%	17.7	+28.3%
	60	6.7	+148.2%	18.1	+31.1%
	40	6.5	+141.0%	19.6	+42.0%
	20	5.4	+100.0%	20.1	+45.7%
300	80	9.4	+147.3%	19.7	+31.3%
	60	10.2	+168.4%	20.7	+38.0%
	40	10.8	+184.2%	22.3	+48.7%
	20	15.8	+315.8%	24.3	+68.0%



(a) The first case (b) The second case

Figure 11. The longitudinal trajectories of HDVs and CAVs.

6. Conclusions

In this paper, a learning scheduling method is proposed for an unsignalized mixed-traffic intersection under the operational condition that reliable explicit HDV turning intention is not available sufficiently early for the scheduling decision. Vehicles are organized into mixed platoons and an HDV group; a state-augmentation function encodes their temporal relationship, and an EN-based expert demonstration stream is incorporated into replay training. The present experimental records support the following observations within the modeled assumptions. The conclusions are intentionally limited to the recorded simulation runs and the tested demand and CAV-penetration scenarios; statistical seed-to-seed robustness, unexpected HDV maneuvers, communication-delay robustness, and hardware-specific deployment latency are outside the empirical scope of the current experiments.

- Without requiring a reliable explicit HDV turn-intention label at the scheduling time, the platoon-level representation can allocate right of way using observable temporal states. HDVs are incorporated either as followers in mixed platoons or into an HDV group, and potentially conflicting scheduling units are temporally separated under the modeled lane-keeping and yielding assumptions.
- In the representative training run, adding EN demonstrations increased the reported peak average reward from approximately −700 to −600 (about 14%). The tested DQN, DDQN, PPO, and SAC implementations achieved lower rewards in the same simulation setup, and the resulting travel costs of 2DQN-Expert were close to the EN benchmark in the reported demand cases. These results are interpreted descriptively for the recorded runs and tested scenarios rather than as statistical significance across independent random seeds.

Future work will relax the conservative information assumptions by fusing turn signal status, lane position, trajectory history, and perception-based probabilistic intention estimates for HDVs. Rather than replacing the present scheduler, such an estimator can provide uncertainty-aware intent probabilities or reachable sets to the coordination layer. Further work should also quantify safety under unexpected HDV maneuvers, evaluate communication/sensing delays, report online inference latency on RSU-class hardware, and validate learning performance across multiple random seeds and CAV penetration rates.

Author Contributions: Conceptualization, X.Y., F.P. and Z.D.; methodology, X.Y. and F.P.; software, X.Y., H.L. and C.S.; validation, X.Y., H.L., C.S. and H.S. (Hao Shi); formal analysis, X.Y. and F.P.; investigation, X.Y., H.L. and H.S. (Hao Shi); resources, Z.D., C.S. and H.S. (Haiming Sun); data curation, X.Y. and H.L.; writing—original draft preparation, X.Y. and F.P.; writing—review and

editing, F.P., Z.D., C.S. and H.S. (Haiming Sun); visualization, X.Y., H.L. and H.S. (Hao Shi); supervision, Z.D.; project administration, Z.D.; funding acquisition, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the Technical Innovation Program of Hubei Province (2024BAB087), in part by the Open Fund Project of State Key Laboratory of Intelligent Green Vehicle and Mobility, (No. KFY260402), in part by the Science and Technology Program of Suzhou (SYG2025116), in part by the Hubei Science and Technology Project (2025CDB015), and in part by the Natural Science Foundation of Jiangsu Province (BK20231197).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Federal Highway Administration. About Intersection Safety. U.S. Department of Transportation. 2026. Available online: https://highways.dot.gov/safety/intersection-safety/about (accessed on 25 September 2026).
2. Wang, X.; Jerome, Z.; Wang, Z.; Zhang, C.; Shen, S.; Kumar, V.V.; Bai, F.; Krajewski, P.; Deneau, D.; Jawad, A.; et al. Traffic light optimization with low penetration rate vehicle trajectory data. *Nat. Commun.* **2024**, *15*, 1306.
3. Oh, S.; Yun, J.; Lee, J. Impact of Mixed Traffic in Urban Networks: Investigating the Effects of Varying Autonomous Vehicle Levels and Penetration Rates. *J. Korean Soc. Transp.* **2023**, *41*, 321–339.
4. Dresner, K.; Stone, P. A multiagent approach to autonomous intersection management. *J. Artif. Intell. Res.* **2008**, *31*, 591–656.
5. Yan, F.; Dridi, M.; Moudni, A.E. Autonomous vehicle sequencing algorithm at isolated intersections. In Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, 4–7 October 2009; pp. 1–6.
6. Carlino, D.; Boyles, S.D.; Stone, P. Auction-based autonomous intersection management. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 529–534.
7. Maguluri, S.T.; Hajek, B.; Srikant, R. The Stability of Longest-Queue-First Scheduling with Variable Packet Sizes. *IEEE Trans. Autom. Control* **2014**, *59*, 2295–2300.
8. Au, T.-C.; Shahidi, N.; Stone, P. Enforcing Liveness in Autonomous Traffic Management. *Proc. AAAI Conf. Artif. Intell.* **2011**, *25*, 1317–1322.
9. Wu, J.; Abbas-Turki, A.; El Moudni, A. Cooperative driving: An ant colony system for autonomous intersection management. *Appl. Intell.* **2012**, *37*, 207–222.
10. Li, Z.; Pourmehrab, M.; Elefteriadou, L.; Ranka, S. Intersection Control Optimization for Automated Vehicles Using Genetic Algorithm. *J. Transp. Eng. Part A Syst.* **2018**, *144*, 04018074.
11. Chouhan, A.P.; Banda, G. Autonomous Intersection Management: A Heuristic Approach. *IEEE Access* **2018**, *6*, 53287–53295.
12. Yao, Z.; Jiang, H.; Jiang, Y.; Ran, B. A Two-Stage Optimization Method for Schedule and Trajectory of CAVs at an Isolated Autonomous Intersection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3263–3281.
13. Xu, H.; Zhang, Y.; Li, L.; Li, W. Cooperative Driving at Unsignalized Intersections Using Tree Search. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4563–4571.
14. Gong, X.; Wang, B.; Liang, S. Collision-Free Cooperative Motion Planning and Decision-Making for Connected and Automated Vehicles at Unsignalized Intersections. *IEEE Trans. Syst. Man. Cybern. Syst.* **2024**, *54*, 2744–2756.
15. Levin, M.W.; Rey, D. Conflict-point formulation of intersection control for autonomous vehicles. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 528–547.
16. Choi, M.; Rubenecia, A.; Choi, H.H. Reservation-based traffic management for autonomous intersection crossing. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 155014771989595.
17. Yu, C.; Sun, W.; Liu, H.X.; Yang, X. Managing connected and automated vehicles at isolated intersections: From reservation- to optimization-based methods. *Transp. Res. Part B Methodol.* **2019**, *122*, 416–435.
18. Chen, X.; Hu, M.; Xu, B.; Bian, Y.; Qin, H. Improved reservation-based method with controllable gap strategy for vehicle coordination at non-signalized intersections. *Phys. A Stat. Mech. Its Appl.* **2022**, *604*, 127953.

19. Li, D.; Wu, J.; Zhu, F.; Chen, T.; Wong, Y.D. Modeling adaptive platoon and reservation-based intersection control for connected and autonomous vehicles employing deep reinforcement learning. *Comput.-Aided Civ. Infrastruct. Eng.* **2023**, *38*, 1346–1364.
20. Zhong, W.; Li, K.; Shi, J.; Yu, J.; Luo, Y. Reservation-Prioritization-Based Mixed-Traffic Cooperative Control at Unsignalized Intersections. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4917–4930.
21. Chen, R.; Hu, J.; Levin, M.W.; Rey, D. Stability-based analysis of autonomous intersection management with pedestrians. *Transp. Res. Part C Emerg. Technol.* **2020**, *114*, 463–483.
22. Yao, Z.; Jiang, H.; Cheng, Y.; Jiang, Y.; Ran, B. Integrated Schedule and Trajectory Optimization for Connected Automated Vehicles in a Conflict Zone. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1841–1851.
23. Pei, H.; Zhang, Y.; Zhang, Y.; Feng, S. Optimal Cooperative Driving at Signal-Free Intersections With Polynomial-Time Complexity. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12908–12920.
24. Jiang, S.; Pan, T.; Zhong, R.; Chen, C.; Li, X.; Wang, S. Coordination of Mixed Platoons and Eco-Driving Strategy for a Signal-Free Intersection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6597–6613.
25. Deng, Z.; Yang, K.; Shen, W.; Shi, Y. Cooperative Platoon Formation of Connected and Autonomous Vehicles: Toward Efficient Merging Coordination at Unsignalized Intersections. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5625–5639.
26. Wu, Y.; Wang, D.Z.W.; Zhu, F. Traffic efficiency and fairness optimisation for autonomous intersection management based on reinforcement learning. *Transp. A Transp. Sci.* **2025**, *21*, 2232047.
27. Zhao, Y.; Gong, D.; Wen, S.; Ding, L.; Guo, G. A Privacy-Preserving-Based Distributed Collaborative Scheme for Connected Autonomous Vehicles at Multi-Lane Signal-Free Intersections. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6824–6835.
28. Cong, X.; Yang, B.; Gao, F.; Chen, C.; Guan, X.; Tang, Y. A Bilevel Virtual Platoon Based Coordination Framework for CAVs at Unsignalized Intersection. *IEEE Trans. Veh. Technol.* **2025**, *74*, 4019–4032.
29. Li, Y.; Liu, M.; Yang, Q.; Shen, Z.; Wu, W. Collision-Free Autonomous Scheduling at Unsignalized Intersection Using Conflict Graph Tree Search. *IEEE Internet Things J.* **2024**, *11*, 14563–14578.
30. Yang, F.; Shen, Y. Distributed Scheduling at Non-Signalized Intersections with Mixed Cooperative and Non-Cooperative Vehicles. *IEEE Trans. Veh. Technol.* **2023**, *72*, 7123–7136.
31. Lioris, J.; Pedarsani, R.; Tascikaraoglu, F.Y.; Varaiya, P. Platoons of connected vehicles can double throughput in urban roads. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 292–305.
32. Wu, R.; Jia, H.; Huang, Q.; Tian, J.; Gao, H.; Wang, G. Multi-Lane Unsignalized Intersection Cooperation Strategy Considering Platoons Formation in a Mixed Connected Automated Vehicles and Connected Human-Driven Vehicles Environment. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 1569–1585.
33. Gong, J.; Zhao, Y.; Cao, J.; Guo, J.; Abdel-Aty, M.; Huang, W. A Virtual Spring Strategy for Cooperative Control of Connected and Automated Vehicles at Signal-Free Intersections. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 1430–1444.
34. Zhao, W.; Liu, R.; Ngoduy, D. A bilevel programming model for autonomous intersection control and trajectory planning. *Transp. A Transp. Sci.* **2021**, *17*, 34–58.
35. Rabiner, L. Combinatorial optimization:Algorithms and complexity. *IEEE Trans. Acoust. Speech Signal Process.* **1984**, *32*, 1258–1259.
36. Lombard, A.; Noubli, A.; Abbas-Turki, A.; Gaud, N.; Galland, S. Deep Reinforcement Learning Approach for V2X Managed Intersections of Connected Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7178–7189.
37. Xu, Y.; Zhou, H.; Ma, T.; Zhao, J.; Qian, B.; Shen, X. Leveraging Multiagent Learning for Automated Vehicles Scheduling at Nonsignalized Intersections. *IEEE Internet Things J.* **2021**, *8*, 11427–11439.
38. Min, J.H.; Ham, S.W.; Kim, D.-K.; Lee, E.H. Deep Multimodal Learning for Traffic Speed Estimation Combining Dedicated Short-Range Communication and Vehicle Detection System Data. *Transp. Res. Rec. J. Transp. Res. Board.* **2023**, *2677*, 247–259.
39. Chen, X.; Jiang, F.J.; Narri, V.; Adnan, M.; Mårtensson, J.; Johansson, K.H. Safe Intersection Coordination with Mixed Traffic: From Estimation to Control. *IFAC-PapersOnLine* **2023**, *56*, 5697–5702.
40. Jiang, J.; Rui, Y.; Ran, B. Unsignalized intersection vehicle platoon formation scheduling method based on mixed traffic. *Sci. Rep.* **2025**, *15*, 26664.
41. Feng, J.; Yuan, X.; Li, Z.; Pan, X.; Liu, K. Right-of-way based multi-agent deep reinforcement learning for collaborative decision-making at unsignalized intersection. *Expert. Syst. Appl.* **2026**, *299*, 130051.
42. Zhang, Q.; Wang, H.; Cai, Y.; Xie, W.-F.; Sun, X.; Chen, L. Stability-constrained coordinated control strategy for vehicle chassis integrated AFS and DYC via reinforcement learning. *Control Eng. Pract.* **2026**, *175*, 107122.
43. Zhou, D.; Hang, P.; Sun, J. Reasoning graph-based reinforcement learning to cooperate mixed connected and autonomous traffic at unsignalized intersections. *Transp. Res. Part C Emerg. Technol.* **2024**, *167*, 104807.

44. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533.
45. Xu, H.; Xiao, W.; Cassandras, C.G.; Zhang, Y.; Li, L. A General Framework for Decentralized Safe Optimal Control of Connected and Automated Vehicles in Multi-Lane Signal-Free Intersections. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17382–17396.
46. Chen, J.; Liang, H.; Li, J.; Lv, Z. Connected Automated Vehicle Platoon Control With Input Saturation and Variable Time Headway Strategy. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4929–4940.
47. Krauss, S. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. Ph.D. Thesis, University of Cologne, Cologne, Germany, 1998; DLR-Forschungsbericht 98-08; Deutsches Zentrum für Luft- und Raumfahrt (DLR): Cologne, Germany, 1998.
48. Zheng, G.; Chen, Y.; Wu, Z.; He, Z.; Zeng, H. Cooperative decision-making in mixed traffic via conflict-aware Heterogeneous Graph Reinforcement Learning. *Simul. Model. Pract. Theory* **2026**, *148*, 103268.
49. Hasselt, H.v.; Guez, A.; Silver, D. Deep reinforcement learning with double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2094–2100.
50. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
51. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
52. Ma, W.; Liao, D.; Liu, Y.; Lo, H.K. Optimization of pedestrian phase patterns and signal timings for isolated intersection. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 502–514.
53. Yu, C.; Ma, W.; Han, K.; Yang, X. Optimization of vehicle and pedestrian signals at isolated intersections. *Transp. Res. Part B Methodol.* **2017**, *98*, 135–153.