EVENT-TRIGGERED SAFE-AWARE PLATOON RE-SEQUENCING FOR SOC BALANCE OF CONNECTED ELECTRIC VEHICLES

Anonymous authors
Paper under double-blind review

ABSTRACT

The position-dependent aerodynamic benefits in connected electric vehicle (EV) platoons provide an opportunity to improve energy efficiency, but they may also result in uneven state-of-charge (SoC) distribution among vehicles during long-distance driving. This paper proposes an event-triggered safe-aware platoon re-sequencing framework to achieve adaptive energy balance optimization while ensuring safe operation in dynamic traffic environments. The platoon re-sequencing problem is formulated by considering the coupling relationships among vehicle positions, aerodynamic energy-saving effects, battery evolution, and traffic-dependent feasibility constraints. To enable safety-aware decision making, a traffic prediction and classification module based on long short-term memory networks and fuzzy c-means clustering is developed to predict future traffic conditions and determine whether platoon re-sequencing can be safely executed. Furthermore, an event-triggered mechanism (ETM) is designed to adaptively regulate the execution of re-sequencing actions according to the real-time SoC imbalance and travel progress of the platoon. Two triggering strategies, including a state-aware self-resetting trigger and an adaptive cooldown-based trigger, are developed to dynamically determine appropriate re-sequencing instants under persistent energy imbalance conditions. The proposed framework separates re-sequencing decision generation from action execution, where a reinforcement learning policy provides candidate platoon orders while the traffic prediction module and ETM jointly determine their safe and adaptive execution. The proposed method enables connected EV platoons to autonomously adjust their formation sequence in response to evolving energy distribution and traffic conditions, thereby improving terminal SoC balance while maintaining safe platoon operation.

1 INTRODUCTION

With the rapid development of connected and autonomous vehicle technologies, vehicle platooning has emerged as a promising approach for improving transportation efficiency and reducing energy consumption Martínez-Díaz et al. (2024); Lesch et al. (2021); Wang et al. (2026). By enabling vehicles to travel with shorter inter-vehicle gaps through vehicle-to-vehicle communication and coordinated control, platooning can effectively reduce aerodynamic drag and achieve considerable energy savings compared with isolated vehicles Liu et al. (2024); Peng et al. (2026); Guo & Meng (2025); Yang et al. (2025). Existing studies have extensively investigated platoon control problems, including longitudinal control Chen et al. (2024); Meng & Cassandras (2020), spacing regulation Lacombe et al. (2023), and energy-efficient coordination Zhu et al. (2022). However, most conventional platoon strategies adopt a fixed formation structure, where each vehicle maintains a predetermined position throughout the entire driving process. Since aerodynamic characteristics vary significantly among different platoon positions, vehicles experience unequal energy-saving benefits Hu et al. (2022). In particular, the leading vehicle suffers from higher aerodynamic resistance due to direct exposure to airflow, whereas following vehicles benefit from the wake effect generated by preceding vehicles Li et al. (2024); Beaver & Malikopoulos (2022); Han et al. (2025). Consequently, fixed platooning inevitably causes heterogeneous energy consumption among vehicles, resulting in imbalanced battery state-of-charge (SoC) distributions and inconsistent remaining driving ranges.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

For electric vehicle (EV) platoons, such energy imbalance becomes a critical issue because the battery state directly affects driving range, battery utilization, and charging coordination Li et al. (2025); Kim et al. (2026). To alleviate this problem, recent studies have explored non-fixed platooning strategies, where vehicles dynamically change their positions to redistribute position-dependent aerodynamic benefits among platoon members. Existing optimization-based approaches have demonstrated that vehicle re-sequencing can effectively reduce terminal SoC imbalance by assigning vehicles to different aerodynamic positions during the trip. For example, Guo & Meng (2025); Guo et al. (2022); Peng et al. (2026) formulated the re-sequencing problem as a minimum-variance optimization problem and developed several algorithms to balance final SoC distributions. However, most existing methods determine vehicle orders based on idealized conditions and focus mainly on optimization performance, while the practical execution of re-sequencing actions is insufficiently considered Guo et al. (2022); Guo & Meng (2025); Zheng et al. (2025); Srisomboon & Lee (2021). In real-world driving scenarios, frequent vehicle position changes may introduce additional communication and control requirements, and inappropriate re-sequencing decisions under unfavorable traffic conditions may compromise platoon safety Selim et al. (2025). Furthermore, conventional optimization methods usually require solving complex combinatorial problems, where the number of possible platoon formations increases exponentially with platoon size and decision opportunities. Therefore, an effective re-sequencing framework should not only generate energy-efficient vehicle orders but also determine when and under what conditions these orders can be safely executed.

To address these challenges, this paper proposes an event-triggered safe-aware platoon re-sequencing framework for connected electric vehicles based on deep reinforcement learning (RL). Different from existing methods that continuously perform optimization or rely on predefined decision intervals Zheng et al. (2025); Liu et al. (2024); Yang et al. (2025), the proposed framework separates the generation of candidate platoon orders from their actual execution. A DQN-based policy is developed to learn energy-efficient re-sequencing strategies by considering the current platoon state and SoC distribution Mnih et al. (2013). Meanwhile, an event-triggered mechanism (ETM) is introduced to adaptively determine suitable re-sequencing instants according to the evolving energy imbalance and travel progress, avoiding unnecessary position adjustments. In addition, a traffic prediction module is integrated to evaluate whether the surrounding environment allows safe execution of re-sequencing actions. The final decision is therefore jointly determined by the learned re-sequencing policy and safety-aware execution constraints, enabling the platoon to dynamically redistribute aerodynamic benefits while maintaining safe operation. Through this framework, the proposed method achieves a balance between energy efficiency, SoC equalization, and practical feasibility of platoon re-sequencing.

The main contributions of this paper are summarized as follows:

1. An event-triggered safe-aware platoon re-sequencing framework is developed to achieve adaptive energy balancing while reducing unnecessary re-sequencing operations. Different from existing re-sequencing methods that optimize vehicle orders at predefined locations or fixed intervals, the proposed framework introduces an ETM to determine appropriate decision instants according to the evolution of platoon states Guo & Meng (2025); Peng et al. (2026); Liu et al. (2024). By avoiding redundant position adjustments while maintaining effective energy redistribution, the proposed framework improves the practicality and efficiency of EV platoon re-sequencing.
2. A safety-aware execution mechanism is proposed to ensure that re-sequencing actions are performed only under feasible traffic conditions. Unlike existing studies that mainly focus on the optimization of vehicle orders under ideal traffic assumptions, the proposed method considers the influence of surrounding traffic environments on the feasibility of vehicle position changes. By integrating traffic state prediction and safety constraints into the decision-making process, unsafe re-sequencing operations caused by dynamic traffic variations can be effectively avoided Guo & Meng (2025); Yang et al. (2025); Zheng & Guo (2025). This provides a more reliable solution for applying platoon re-sequencing in real-world driving scenarios.
3. A deep RL-based re-sequencing policy is designed to efficiently generate energy-oriented platoon orders. Compared with conventional combinatorial optimization methods that suffer from rapidly increasing computational complexity with platoon size and decision opportunities, the proposed DQN-based policy learns the relationship between platoon states and

optimal position assignments through interactions with the environment Guo et al. (2022). The learned policy can dynamically exploit position-dependent aerodynamic benefits to balance terminal SoC distribution while maintaining computational efficiency. Therefore, the proposed approach provides an intelligent and scalable solution for adaptive EV platoon energy management.

2 PROBLEM FORMULATION

To achieve balanced energy consumption within an EV platoon in dynamic traffic environments, this study formulates an event-triggered safe-aware platoon re-sequencing problem. The objective is to redistribute position-dependent energy-saving benefits among EVs while ensuring that re-sequencing maneuvers are performed only under safe and feasible traffic conditions. By explicitly considering both energy imbalance and re-sequencing safety, the formulated problem provides a practical basis for dynamic platoon coordination in real-world traffic environments.

Consider a platoon consisting of N connected EVs, where $N \geq 2$ denotes the total number of vehicles in the platoon. The vehicle set is denoted by $\Omega \triangleq \{1, 2, \ldots, N\}$. Here, Ω represents the set of all vehicle indices, and $i \in \Omega$ denotes the index of an individual vehicle. The platoon re-sequencing decision is made at a sequence of decision instants. Let k denote the index of the re-sequencing decision instant, and let t_k denote the physical time corresponding to the k-th decision instant. Different from a fixed-period formulation, t_k is not necessarily equally spaced in time because the decision instants are generated by an ETM. At decision instant t_k, the position assigned to vehicle i is denoted by $P(i,k) \in \{1, 2, \ldots, N\}$. The symbol $P(i,k)$ is a position assignment variable. It represents the position occupied by vehicle i in the platoon at the k-th decision instant. Specifically, $P(i,k) = 1$ means that vehicle i is assigned to the leading position of the platoon. The complete platoon order at decision instant t_k is represented by $P(k) = [P(1,k), P(2,k), \ldots, P(N,k)]^T$. The vector $P(k)$ collects the position assignments of all vehicles and therefore describes the entire ordering of the platoon at t_k. In the re-sequencing problem, $P(k)$ is the decision variable to be selected at each event-triggered decision instant. Together with the definition $P(i,k)$, the one-to-one assignment condition ensures that $P(k)$ is a valid permutation of the N platoon positions. The set of all feasible platoon orders is denoted by $\mathcal{P}$. Since the N vehicles can be assigned to the N platoon positions in $N!$ different permutations, the number of feasible platoon orders satisfies $|\mathcal{P}| = N!$. Each element of $\mathcal{P}$ corresponds to one admissible ordering of the N vehicles. Therefore, selecting a re-sequencing decision at t_k is equivalent to choosing one feasible order $P(k) \in \mathcal{P}$.

Let $s_i(t)$ denote the SoC of vehicle i at time t. The SoC $s_i(t)$ represents the remaining battery energy level of vehicle i. The SoC vector of the entire platoon is defined as $s(t) = [s_1(t), s_2(t), \ldots, s_N(t)]^T$. The vector $s(t)$ describes the energy distribution among all vehicles in the platoon at time t. The purpose of re-sequencing is to make the $s(t)$ more balanced throughout the trip. The energy consumption of each vehicle depends on its position in the platoon. Let $e_j, j \in \{1, 2, \ldots, N\}$ denote the energy-saving coefficient associated with the j-th platoon position.The index j represents a physical platoon position rather than a vehicle index. A larger value of e_j means that the vehicle located at position j obtains a greater aerodynamic drag-reduction benefit and therefore consumes less energy over the same traveled distance. If vehicle i is assigned to position $P(i,k)$ during the interval $[t_k, t_{k+1})$, then the energy-saving coefficient applied to vehicle i in that interval is $e_{P(i,k)}$. This notation links the re-sequencing decision $P(i,k)$ with the energy consumption model. Once the position of vehicle i changes, the corresponding energy-saving coefficient also changes. Let δ denote the energy consumption per unit distance when no position-dependent aerodynamic savings are considered. Let Λ denote the battery capacity of each EV. The ratio δ/Λ converts distance-dependent energy consumption into the corresponding SoC reduction. Let $d_i(t)$ denote the traveled distance of vehicle i up to time t. Therefore, $d_i(t_{k+1}) - d_i(t_k)$ represents the distance traveled by vehicle i during the interval $[t_k, t_{k+1})$. The SoC dynamics of vehicle i over the interval $[t_k, t_{k+1})$ are given by

$$s_i(t_{k+1}) = s_i(t_k) - \frac{\delta}{\Lambda}(1 - e_{P(i,k)})[d_i(t_{k+1}) - d_i(t_k)], i \in \Omega. \tag{1}$$

In this equation, $s_i(t_k)$ is the initial SoC of vehicle i at the beginning of the decision interval, and $s_i(t_{k+1})$ is the SoC after traveling to the next decision instant. The term $\frac{\delta}{\Lambda}[d_i(t_{k+1}) - d_i(t_k)]$ represents the SoC reduction over the traveled distance, while $1 - e_{P(i,k)}$ adjusts this consump-

tion according to the platoon position assigned to vehicle i. Therefore, the re-sequencing decision directly affects the SoC evolution of each vehicle through the position-dependent coefficient $e_{P(i,k)}$.

To ensure that platoon re-sequencing is feasible in dynamic traffic environments, the surrounding traffic conditions are modeled. The traffic state is used to determine whether position switching can be safely performed at a decision instant. The traffic condition of a road segment is characterized by a traffic characteristic vector $x = [\mathcal{F}, \mathcal{V}, \mathcal{K}]$. In this vector, x denotes the traffic feature vector used to describe the traffic environment. The three components $\mathcal{F}$, $\mathcal{V}$ and $\mathcal{K}$ represent traffic flow, average vehicle speed, and traffic density, respectively. The symbol $\mathcal{F}$ represents the number of vehicles passing through a fixed cross-section of the road segment per unit of time. A larger value of $\mathcal{F}$ means that more vehicles pass through the considered road segment within the same time interval. The symbol $\mathcal{V}$ is measured as the space-mean speed of vehicles traveling on the segment during the aggregation period. A smaller value of $\mathcal{V}$ usually indicates slower traffic movement and may imply more restrictive traffic conditions. The symbol $\mathcal{K}$ represents the number of vehicles present per unit length of the roadway. A larger value of $\mathcal{K}$ indicates that vehicles are more densely distributed on the road, which may reduce the available space for safe position switching. The relationship among traffic flow, average speed, and density is expressed as $\mathcal{K} = \mathcal{F}/\mathcal{V}$. This equation indicates that, for a given traffic flow, a lower average speed corresponds to a higher density. Since dense traffic may restrict position-switching maneuvers, $\mathcal{K}$ is an important indicator for judging re-sequencing feasibility. In addition, a lane-change permission indicator is introduced to describe whether the road regulations of the current segment permit platoon re-sequencing. It is defined as

$$\ell(t_k) = \begin{cases} 1, & \text{if platoon re-sequencing is permitted,} \\ 0, & \text{otherwise.} \end{cases}$$

Because the re-sequencing decision should account for near-future traffic conditions rather than only the current observation, predicted traffic information is considered. Let $x^{(k)} = [x_t, x_{t-1}, \ldots, x_{t-k+1}]^T$ denote the historical traffic indicator vector with k time steps. The predicted traffic indicator vector for the next time step is denoted by $\hat{x}_{t+1}$. Based on the predicted traffic indicators, the safety-related traffic state at decision instant t_k is denoted by $\rho(t_k) \in \mathcal{C}$. The symbol $\rho(t_k)$ represents the traffic condition category used for safety-aware re-sequencing decisions. It describes whether the traffic environment at decision instant t_k is suitable for position switching. The set $\mathcal{C} = \{1, 2, \ldots, g\}$ denotes the finite set of traffic condition categories, where g is the total number of categories. A larger value of $\rho(t_k)$ can be used to represent a more unsafe traffic condition. Let ρ_{th} denote the safety threshold for re-sequencing feasibility. This threshold separates traffic states in which re-sequencing is allowed from traffic states in which re-sequencing is prohibited. If $\rho(t_k) \geq \rho_{th}$, the traffic condition is considered unsafe for re-sequencing. In this case, the platoon must keep its previous order, which is expressed as

$$P(i,k) = P(i,k-1), \forall i \in \Omega.$$

Here, $P(i, k-1)$ denotes the position of vehicle i at the previous decision instant. This constraint means that when the predicted traffic state is above the safety threshold, no position reassignment is permitted. Therefore, the safety-aware traffic state acts as a feasibility constraint on the platoon re-sequencing decision.

Let s_{min} denote the minimum allowable SoC. This lower bound represents the minimum battery level that a vehicle is allowed to reach during the trip. The terminal time of the trip is defined as

$$t_f = \inf\{t | s_i(t) = s_{min}, \exists i \in \Omega\}. \tag{2}$$

Here, t_f denotes the first time at which at least one vehicle reaches the minimum allowable SoC. The notation $\exists i \in \Omega$ means that there exists at least one vehicle in the platoon whose SoC reaches s_{min}. This terminal condition reflects the fact that the platoon operation is limited by the vehicle with the lowest remaining energy. The terminal SoC vector is

$$s(t_f) = [s_1(t_f), s_2(t_f), \ldots, s_N(t_f)]^T.$$

This vector contains the SoC values of all vehicles at terminal time. The terminal mean SoC is defined as

$$\bar{s}(t_f) = \frac{1}{N} \sum_{i=1}^{N} s_i(t_f).$$

The terminal SoC standard deviation is defined as

$$\sigma_s(t_f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i(t_f) - \bar{s}(t_f))^2}.$$

The symbol $\sigma_s(t_f)$ measures the terminal SoC dispersion among vehicles. A smaller value of $\sigma_s(t_f)$ indicates that the remaining energy is more evenly distributed across the platoon at the end of the trip. Therefore, minimizing $\sigma_s(t_f)$ is equivalent to improving the final energy balance among EVs.

3 METHODS

3.1 TRAFFIC STATE PREDICTION AND CLASSIFICATION

The safe-aware execution layer predicts the near-future traffic condition before a candidate platoon order is applied. Let

$$\mathbf{x}_t = [F_t, V_t, K_t]$$

denote the traffic feature vector containing flow, average speed, and density. Given a historical observation window $x^{(k)}$, an LSTM model predicts the next traffic state:

$$\hat{\mathbf{x}}_{t_k+1} = f_{LSTM}\big(\mathbf{x}^{(k)}; \theta_{LSTM}\big).$$

The predicted vector is subsequently classified by FCM into four traffic categories corresponding to free flow, basic free flow, mild congestion, and severe congestion. Re-sequencing is permitted only when

$$\rho(t_k) \in \{1, 2\}, \ell(t_k) = 1,$$

where $\ell(t_k)$ denotes the lane-change permission indicator. Otherwise, the previous platoon order is retained. Detailed LSTM training and FCM clustering procedures are provided in Appendix A.

3.2 EVENT-TRIGGERED PLATOON RE-SEQUENCING

The traffic-state prediction module determines whether the current traffic condition permits platoon re-sequencing. However, even under feasible traffic conditions, updating the platoon order at every decision instant may introduce excessive communication and actuation costs. Therefore, two ETMs are introduced to regulate the execution of the platoon re-sequencing.

At each decision instant t_k, the SoC dispersion is measured by

$$V(t_k) = \frac{1}{N}\sum_{i=1}^{N}(s_i(t_k) - \bar{s}(t_k))^2. \tag{3}$$

To describe the trip progress, the platoon-level cumulative travel distance is defined as

$$\bar{d}(t_k) = \frac{1}{N}\sum_{i=1}^{N} d_i(t_k). \tag{4}$$

Let $d_{\max} = \Lambda s_0/\delta$ denote the maximum travel distance that a vehicle can reach without considering aerodynamic drag. The normalized cumulative travel distance is then defined as

$$d_k^{\text{norm}} = \frac{\bar{d}(t_k)}{d_{\max}}. \tag{5}$$

Based on d_k^{norm}, the distance-adaptive triggering threshold is designed as

$$\Gamma(t_k) = \epsilon \bar{s}^2(t_k)\max(0, 1 - \beta d_k^{\text{norm}}) + \sigma^2, \tag{6}$$

where $\Gamma(t_k)$ is the upper triggering threshold at decision instant t_k, and $\epsilon > 0$, $\beta > 0$, and $\sigma > 0$ are design parameters.

The design of equation 6 adapts the triggering threshold to both the current battery condition and the travel progress of the platoon. The factor $\bar{s}^2(t_k)$ makes the threshold SoC-aware. When the average

SoC is relatively high, the platoon can tolerate a larger SoC dispersion, which reduces unnecessary re-sequencing operations. As the average SoC decreases, the admissible imbalance becomes smaller, making the event trigger more sensitive to SoC deviation. The term $\max(0, 1 - \beta d_k^{\mathrm{norm}})$ tightens the threshold as the cumulative travel distance increases. Since $d_k^{\mathrm{norm}} = \bar{d}(t_k)/d_{\max}$, a larger d_k^{norm} indicates that the platoon is closer to the maximum reachable distance under the no-aerodynamic-drag assumption. Therefore, the remaining margin for correcting accumulated SoC imbalance becomes smaller, and the re-sequencing mechanism should impose a stricter requirement on SoC consistency. The residual term σ^2 prevents the threshold from collapsing to zero and preserves a positive triggering margin.

The ETM acts as an execution gate for the candidate platoon order generated by the DRL policy. Let $\pi_\theta(\cdot)$ denote the learned DRL policy parameterized by θ. The executed platoon order is determined by

$$P(k) = \begin{cases} \pi_\theta(x_k), & \rho(t_k) \in \{1, 2\} \wedge \ell(t_k) = 1 \wedge \mathrm{ETM}(t_k) = 1, \\ P(k-1), & \text{otherwise}, \end{cases} \tag{7}$$

where x_k denotes the DRL state at decision instant t_k, and $\mathrm{ETM}(t_k) \in \{0, 1\}$ is the event-trigger indicator. If the predicted traffic condition is feasible and the event condition is satisfied, the candidate platoon order is executed. Otherwise, the platoon maintains the previous order.

Two event-triggered mechanisms are considered. ETM_1 is a state-aware self-resetting trigger designed for persistent SoC imbalance, whereas ETM_2 is an adaptive cooldown-based trigger designed to explicitly impose a dwell time between consecutive re-sequencing events.

3.2.1 ETM_1: State-Aware Self-Resetting Trigger

ETM_1 is designed as a state-aware self-resetting ETM. Unlike a conventional hysteresis trigger, which is re-armed only after the monitored variable falls below a lower threshold, ETM_1 allows re-arming while the system remains in a persistently imbalanced regime. This feature is suitable for long-distance EV platooning, where a single re-sequencing action may not immediately reduce SoC dispersion.

Based on equation 6, the self-resetting threshold is defined as

$$\underline{\Gamma}(t_k) = \rho_e \Gamma(t_k), 0 < \rho_e < 1, \tag{8}$$

where $\underline{\Gamma}(t_k)$ is the lower self-resetting threshold and ρ_e is the threshold ratio. Let $q_k \in \{0, 1\}$ denote the internal trigger state at decision instant t_k, where $q_k = 1$ indicates that the trigger is armed and $q_k = 0$ indicates that the trigger is disarmed. The self-resetting rule is

$$q_k \leftarrow 1, \quad \text{if} \quad q_k = 0 \wedge V(t_k) \geq \underline{\Gamma}(t_k). \tag{9}$$

After the possible re-arming operation, the triggering condition is checked as

$$\mathrm{ETM}_1(t_k) = \begin{cases} 1, & q_k = 1 \wedge V(t_k) \geq \Gamma(t_k), \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

If $\mathrm{ETM}_1(t_k) = 1$, a re-sequencing event is generated and the trigger state is reset as $q_{k+1} = 0$. Otherwise, q_{k+1} follows the updated value after the re-arming rule in equation 9.

Algorithm 1: ETM_1**-Based Re-Sequencing Gate** At decision instant t_k, compute $V(t_k)$, $\bar{d}(t_k)$, d_k^{norm}, $\Gamma(t_k)$, and $\underline{\Gamma}(t_k)$. Initialize $\mathrm{ETM}_1(t_k) = 0$. If $\rho(t_k) \notin \{1, 2\}$ or $\ell(t_k) = 0$, set $P(k) = P(k-1)$, keep $q_{k+1} = q_k$, and terminate the update. Otherwise, if $q_k = 0$ and $V(t_k) \geq \underline{\Gamma}(t_k)$, set $q_k = 1$. Then, if $q_k = 1$ and $V(t_k) \geq \Gamma(t_k)$, set $\mathrm{ETM}_1(t_k) = 1$, execute $P(k) = \pi_\theta(x_k)$, and reset $q_{k+1} = 0$. If this condition is not satisfied, set $P(k) = P(k-1)$ and $q_{k+1} = q_k$. The Appendix B verifies that ETM_1 is Zeno-free in the proposed re-sequencing framework.

It is worth noting that the auxiliary margin does not require $V(t)$ to fall below the lower threshold before ETM_1 can be reactivated. As $\eta(t)$ is gradually released after each event, the triggering condition can become active when a persistent SoC imbalance remains. Therefore, the proposed construction preserves the self-resetting capability of ETM_1 while guaranteeing Zeno-freeness.

3.2.2 ETM_2: ADAPTIVE COOLDOWN-BASED TRIGGER

ETM_2 explicitly imposes a positive dwell time after each re-sequencing event. Let $C_k \in \mathbb{Z}_{\geq 0}$ denote the cooldown counter at decision instant t_k. When $C_k > 0$, the trigger is locked and no new event is allowed. When $C_k = 0$, the event condition is checked using $\Gamma(t_k)$. The ETM_2 triggering rule is

$$\mathrm{ETM}_2 = \begin{cases} 0, & C_k > 0, \\ 1, & C_k = 0 \wedge V(t_k) \geq \Gamma(t_k), \\ 0, & C_k = 0 \wedge V(t_k) < \Gamma(t_k). \end{cases} \tag{11}$$

If $\mathrm{ETM}_2(t_k) = 1$, the cooldown counter is reset according to the current imbalance severity. Define

$$r_k = \frac{\Gamma(t_k)}{V(t_k)}, 0 < r_k \leq 1, \tag{12}$$

where r_k is the severity ratio. The post-trigger cooldown length is assigned by

$$C_{k+1} = \max(C_{\min}, \lfloor C_{\text{base}} r_k \rfloor), \tag{13}$$

where $C_{\min} \geq 1$ is the minimum cooldown length, $C_{\text{base}} > 0$ is the base cooldown parameter, and $\lfloor \cdot \rfloor$ denotes the floor operator. If $C_k > 0$, then $C_{k+1} = C_k - 1$. If $C_k = 0$ and no event is triggered, then $C_{k+1} = 0$.

Algorithm 2: ETM_2**-Based Re-Sequencing Gate** At decision instant t_k, compute $V(t_k)$, $\bar{d}(t_k)$, d_k^{norm}, and $\Gamma(t_k)$. Initialize $\mathrm{ETM}_2(t_k) = 0$. If $\rho(t_k) \notin \{1, 2\}$ or $\ell(t_k) = 0$, set $P(k) = P(k-1)$, keep $C_{k+1} = C_k$, and terminate the update. Otherwise, if $C_k > 0$, set $P(k) = P(k-1)$, update $C_{k+1} = C_k - 1$, and terminate the update. If $C_k = 0$ and $V(t_k) \geq \Gamma(t_k)$, set $\mathrm{ETM}_2(t_k) = 1$, compute $r_k = \Gamma(t_k)/V(t_k)$, execute $P(k) = \pi_\theta(x_k)$, and set $C_{k+1} = \max(C_{\min}, \lfloor C_{\text{base}} r_k \rfloor)$. Otherwise, set $P(k) = P(k-1)$ and $C_{k+1} = 0$. The Appendix C analysis verifies that ETM_2 is Zeno-free in the proposed discrete-time re-sequencing framework.

Remark 1 *The proposed ETMs do not replace the DRL policy. Instead, they act as an execution gate between the learned policy and the platoon environment. At each decision instant t_k, the DRL policy $\pi_\theta(x_k)$ provides a candidate platoon order. This candidate order is executed only when the predicted traffic category satisfies $\rho(t_k) \in \{1, 2\}$, the lane-change permission indicator satisfies $\ell(t_k) = 1$, and the event condition satisfies $\mathrm{ETM}(t_k) = 1$. Otherwise, the platoon maintains $P(k-1)$.*

This design decouples policy learning from action execution. The DRL agent determines which platoon order is beneficial for SoC balancing, whereas the LSTM-FCM module and the ETM jointly determine when the candidate order should be applied. Therefore, ETM1 and ETM2 are algorithm-agnostic and can be incorporated into different DRL frameworks, including value-based, policy-gradient, and actor-critic methods, as long as the learned policy outputs a candidate platoon order.

3.3 DQN-BASED CANDIDATE PLATOON RE-SEQUENCING POLICY

To determine an energy-efficient platoon order whenever a re-sequencing opportunity becomes available, DQN is employed as the candidate-order generator. The platoon re-sequencing process is modeled as a finite-action MDP, in which the agent learns the long-term influence of different platoon formations on the evolution of the SoC distribution. In contrast to directly embedding traffic feasibility into the learned policy, the proposed architecture separates order optimization from action execution. Specifically, the DQN determines a candidate formation according to the energy state of the platoon, whereas the traffic-state prediction module and the ETM determine whether this candidate formation is admissible for execution. This separation prevents the value network from simultaneously learning energy-balancing behavior and traffic-safety constraints, and allows the same learned policy to be combined with different ETMs.

3.3.1 STATE REPRESENTATION

At decision instant t_k, the DQN should contain sufficient information to characterize the current energy distribution, the existing platoon configuration, and the remaining progress of the trip. Accordingly, the state supplied to the value network is defined as

$$\zeta_k = [s(t_k), P(t_k), d_k^{\text{norm}}], \tag{14}$$

where $s(t_k)$ is the platoon SoC vector, $P(t_k)$ denotes the platoon order currently being executed, and d_k^{norm} is the normalized cumulative travel distance defined previously.

Including the current formation is important because the subsequent SoC evolution depends explicitly on the position-dependent coefficient $e_{P(i,k)}$. Consequently, two platoons with identical SoC vectors but different vehicle orders do not, in general, have identical state transitions. The normalized distance further provides the value network with information regarding the remaining opportunity for correcting accumulated SoC imbalance as the platoon approaches the end of its trip.

It should be emphasized that $\rho(t_k)$, $\ell(t_k)$ and $\mathrm{ETM}(t_k)$ are not included in the DQN state. These quantities characterize whether a candidate action can be safely and adaptively executed rather than which platoon order is energetically desirable. This treatment is consistent with the separation between policy generation and action execution adopted in the proposed framework.

3.3.2 Discrete Candidate-Order Space

For a platoon consisting of N EVs, the feasible platoon-order set has already been defined as $\mathcal{P}$, with $|\mathcal{P}| = N!$. Each admissible permutation is treated as one discrete DQN action. Hence, the action space can be written as

$$\mathcal{A} \equiv \mathcal{P}. \tag{15}$$

Rather than directly overwriting the executed order $P(t_k)$, the output of the DQN is defined as a candidate platoon order

$$\widetilde{P}(t_k) = \pi_\theta(\zeta(t_k)),$$

where θ denotes the parameters of the Q-network. The deterministic greedy policy associated with the DQN is

$$\pi_\theta(\zeta(t_k)) = \arg\max_{P' \in \mathcal{P}} Q(\zeta(t_k), P'; \theta), \tag{16}$$

where $Q(\zeta(t_k), P'; \theta)$ approximates the expected discounted return associated with selecting candidate order P' from state $\zeta(t_k)$.

During training, an ε-greedy strategy is adopted to balance exploration and exploitation:

$$\widetilde{P}(t_k) = \begin{cases} P_{\text{rand}}, & \xi(t_k) < \varepsilon(t_k), \\ \arg\max_{P' \in \mathcal{P}} Q(\zeta(t_k), P'; \theta) & \xi(t_k) \geq \varepsilon(t_k), \end{cases}$$

where $\xi(t_k) \sim \mathcal{U}(0,1)$, P_{rand} is uniformly sampled from $\mathcal{P}$, and $\varepsilon(t_k)$ is gradually reduced during training. After convergence, exploration is disabled and the greedy candidate order in equation 16 is used. The distinction between $\widetilde{P}(t_k)$ and $P(t_k)$ is essential. The former denotes the formation preferred by the DQN from the perspective of energy balancing, whereas the latter represents the formation actually applied to the platoon.

3.3.3 Energy-Balance-Oriented Reward

The learning objective of the DQN is required to remain consistent with the terminal SoC-balancing objective formulated in Section 2. Directly using only $\sigma_s(t_f)$ as a terminal reward, however, produces a sparse learning signal and makes credit assignment difficult for long-distance platooning tasks. To provide informative feedback during the trip, the instantaneous change in the SoC-dispersion function $V(t_k)$ is therefore used for reward shaping.

For a nonterminal transition, the reward is defined as

$$r(t_k) = \omega_b \frac{V(t_{k-1}) - V(t_k)}{V(t_{k-1})}, t_k < t_f, \tag{17}$$

where $\omega_b > 0$ is a balancing-reward coefficient. A reduction in dispersion therefore yields a positive reward, whereas a formation that aggravates the energy imbalance receives a negative reward. At the terminal time t_f, an additional terminal term is introduced:

$$r(t_f) = \omega_b \frac{V(t_{f-1}) - V(t_f)}{V(t_{f-1})} - \omega_f \sigma_s(t_f), \tag{18}$$

Since t_f is defined as the first instant at which one of the EVs reaches $s_{\min}$, the second term directly preserves the optimization objective of minimizing the terminal SoC standard deviation. The

normalized dispersion-improvement term provides dense feedback regarding whether a candidate formation is moving the platoon toward a more balanced energy state, while the terminal penalty prevents the learning process from favoring short-term reductions in dispersion at the expense of the final energy distribution.

Importantly, no additional penalty associated with $\rho(t_k)$, $\ell(t_k)$, or triggering frequency is embedded in the DQN reward. Traffic feasibility and event frequency are handled explicitly by the safety and ETM layers. Consequently, the learned Q-function remains dedicated to evaluating the long-term energy consequence of candidate platoon formations rather than implicitly approximating constraints that are already enforced by the execution mechanism. The candidate-order policy is trained using standard DQN with experience replay and a target network. The complete optimization procedure and hyperparameters are reported in Appendix D. The complete illustration of the proposed algorithm is provided in Appendix E.

4 Experiment

4.1 Experimental design

Experiments are conducted with platoons containing $4 \sim 7$ EVs. Each vehicle starts from a fully charged state, with battery capacity $\Lambda = 85$ kWh and baseline energy-consumption rate $\delta = 211$ Wh/km. The position-dependent aerodynamic saving coefficients are $e_1 = 0.043$, $e_2 = 0.10$, and $e_i = 0.14$ for $i \geq 3$. All compared algorithms are evaluated under identical vehicle parameters, traffic scenarios, and termination conditions. Detailed neural-network configurations, ETM parameters, traffic-data settings, and training hyperparameters are provided in Appendix F.

4.2 Experimental Results and Discussion

The evaluated re-sequencing policies are compared under both event-triggering mechanisms for platoons containing four to seven EVs. Complete results for episode reward, total travel distance, and re-sequencing frequency are reported in Appendix G. Under ETM_1, DQN achieves an average episode reward of -6.245 with 3.25 re-sequencing operations, providing a relatively stable trade-off between decision quality and execution frequency across different platoon sizes.

A more pronounced improvement is observed under ETM_2. For DQN, the average episode reward increases from -6.245 to -2.238, while the average travel distance increases from 271.12 km to 278.58 km. This improvement is accompanied by an increase in the average number of re-sequencing operations from 3.25 to 4.50. Thus, the advantage of ETM_2 lies not in minimizing the number of events, but in regulating their timing to provide more effective opportunities for correcting accumulated SoC imbalance.

Although no candidate-order policy dominates every platoon size, DQN provides a favorable overall trade-off among reward, travel distance, and re-sequencing frequency. Moreover, its discrete action formulation naturally matches the finite platoon-permutation space. Therefore, DQN is adopted as the candidate-order generator, while ETM_2 is selected as the primary execution mechanism.

5 Conclusion

This paper investigated event-triggered safe-aware re-sequencing for SoC balancing in connected EV platoons. By separating candidate-order generation from execution, the proposed framework integrates a DQN policy with traffic-feasibility constraints and two SoC-aware triggering mechanisms. Simulation results show that the adaptive cooldown mechanism achieves higher episode rewards than the self-resetting mechanism across the evaluated platoon sizes, although it generally requires more re-sequencing operations. These findings emphasize the importance of coordinating formation selection with adaptive execution timing. Future work will investigate prediction uncertainty, explicit lane-changing dynamics, and larger platoon configurations to further assess the framework under realistic traffic conditions.

AI USE STATEMENT

In this work, generative AI tools were used to polish and refine the English writing of the manuscript, and to assist with literature retrieval and discovery of related references. We have not used generative AI tools for research ideation, algorithm design, simulation implementation, mathematical proof, dataset generation, or drafting core sections of this paper. All AI-assisted text and retrieved materials were manually reviewed, fact-checked, and revised by the authors. The authors take full responsibility for all content, claims, and results presented in this paper.

ETHICS STATEMENT

This study involves no human participants or personally identifiable information. The proposed safety-aware mechanism is evaluated only in simulation and should not be interpreted as a certified safety guarantee for real-world vehicle operation.

REPRODUCIBILITY STATEMENT

The environment configuration, reward formulation, event-triggered rules and hyperparameters are fully described in the main text and appendix. Readers can reproduce the experiments following the detailed descriptions provided in this paper.

REFERENCES

Logan E. Beaver and Andreas A. Malikopoulos. Constraint-Driven Optimal Control of Multiagent Systems: A Highway Platooning Case Study. *IEEE Control Systems Letters*, 6:1754–1759, 2022. doi: 10.1109/LCSYS.2021.3133801.

Xiang Chen, Shuo Cheng, Shaohua Li, Wanzhong Zhao, and Chunyan Wang. Lateral and Longitudinal Integrated Emergency Collision Avoidance System with Handling Stability Guarantee. *IEEE Transactions on Transportation Electrification*, 10(3):4930–4940, 2024. doi: 10.1109/TTE.2023.3323643.

Shaopan Guo and Xiangyu Meng. Optimal Resequencing of Connected and Autonomous Electric Vehicles in Battery SOC-Aware Platooning. *IEEE Transactions on Transportation Electrification*, 11(4):9298–9305, 2025. doi: 10.1109/TTE.2025.3561311.

Shaopan Guo, Xiangyu Meng, and Mehdi Farasat. Energy Efficient and Battery SOC-aware Coordinated Control of Connected and Autonomous Electric Vehicles. In *2022 American Control Conference (ACC)*, pp. 4707–4712, 2022. doi: 10.23919/ACC53348.2022.9867292.

Qiaoni Han, Jianguo Ma, Zhiqiang Zuo, Xiaocheng Wang, and Xinping Guan. Data-Driven Finite-Time Platooning Control for Heterogeneous Nonlinear Vehicle Systems. *IEEE Control Systems Letters*, 9:33–37, 2025. doi: 10.1109/LCSYS.2025.3552676.

Manjiang Hu, Chongkang Li, Yougang Bian, Hui Zhang, Zhaobo Qin, and Biao Xu. Fuel Economy-Oriented Vehicle Platoon Control Using Economic Model Predictive Control. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20836–20849, 2022. doi: 10.1109/TITS.2022.3183090.

Eunwoo Kim, Hongkyun Park, Nahyun Lee, Kil Young Lee, Sangyoung Park, and Jong-Chan Kim. E-Platoon: e-Truck Platoon Simulator Based on CARLA Towards Platoon-level Battery Management. In *2026 IEEE Intelligent Vehicles Symposium (IV)*, pp. 787–794, 2026. doi: 10.1109/IV66570.2026.11623876.

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár. Distributed Eco-Driving Control of a Platoon of Electric Vehicles Through Riccati Recursion. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3048–3063, 2023. doi: 10.1109/TITS.2022.3224389.

Veronika Lesch, Christian Krupitzer, Kevin Stubenrauch, Nico Keil, Christian Becker, Samuel Kounev, and Michele Segata. A Comparison of Mechanisms for Compensating Negative Impacts of System Integration. *Future Generation Computer Systems*, 116:117–131, 2021.

ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2020.10.029. URL https://www.sciencedirect.com/science/article/pii/S0167739X20330132.

Meng Li, Zehong Cao, and Zhibin Li. Augmented mixed vehicular platoon control with dense communication reinforcement learning for traffic oscillation alleviation. *IEEE Internet of Things Journal*, 11(22):35989–36001, 2024. doi: 10.1109/JIOT.2024.3409618.

Zhicheng Li, Huawei Niu, Haoyu Miao, and Yang Wang. Multi-Object Optimization of Battery Management for Electric Vehicle Platooning Considering Energy Consumption and Battery Health. *IET Intelligent Transport Systems*, 19(1):e70074, 2025. doi: https://doi.org/10.1049/itr2.70074. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.70074.

Miao Liu, Chu Peng, Shaopan Guo, Long Xiao, Benyun Shi, and Yue Peng. Optimal Re-Sequencing of Electric Vehicle Platoons Based on Deep Reinforcement Learning. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4250–4255, 2024. doi: 10.1109/SMC54092.2024.10831504.

Margarita Martínez-Díaz, Christelle Al-Haddad, Francesc Soriguera, and Constantinos Antoniou. Impacts of Platooning of Connected Automated Vehicles on Highways. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):6366–6396, 2024. doi: 10.1109/TITS.2024.3350776.

Xiangyu Meng and Christos G. Cassandras. Trajectory Optimization of Autonomous Agents with Spatio-Temporal Constraints. *IEEE Transactions on Control of Network Systems*, 7(3):1571–1581, 2020. doi: 10.1109/TCNS.2020.2988005.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, 2013. URL https://arxiv.org/abs/1312.5602.

Chu Peng, Shaopan Guo, Miao Liu, and Long Xiao. Congestion-Aware Platoon Resequencing Optimization for Electric Vehicles Using Deep Reinforcement Learning. *Neurocomputing*, 676:133040, 2026. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2026.133040. URL https://www.sciencedirect.com/science/article/pii/S0925231226004376.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization, 2017. URL https://arxiv.org/abs/1502.05477.

Mahmoud Selim, Sriharsha Bhat, and Karl Henrik Johansson. Bayesian learning of adaptive koopman operator with application to robust motion planning for autonomous trucks, 2025. URL https://openreview.net/forum?id=yIdCQFvbYe.

Issaree Srisomboon and Sanghwan Lee. Efficient Position Change Algorithms for Prolonging Driving Range of a Truck Platoon. *Applied Sciences*, 11(22), 2021. ISSN 2076-3417. doi: 10.3390/app112210516. URL https://www.mdpi.com/2076-3417/11/22/10516.

Yuanming Wang, Xiaoyu Wang, and Shaopan Guo. Truck Platooning via Zeno-Free Event-Triggered Communication Based on Reinforcement Learning. *Electronics*, 15(18), 2026. ISSN 2079-9292. doi: 10.3390/electronics15184165. URL https://www.mdpi.com/2079-9292/15/18/4165.

Zhikai Yang, Shaopan Guo, Xiaoyu Wang, Miao Liu, and Long Xiao. Periodic Selection Reordering Algorithm for Extending Truck Ranking Driving Mileage. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 5242–5247, 2025. doi: 10.1109/SMC58881.2025.11343256.

Baiwenjie Zheng and Shaopan Guo. Dynamic Platoon Re-Sequencing for Electric Vehicles Based on Bootstrapped DQN. *Electronics*, 14(22), 2025. ISSN 2079-9292. doi: 10.3390/electronics14224417. URL https://www.mdpi.com/2079-9292/14/22/4417.

BaiWenjie Zheng, Shaopan Guo, Miao Liu, and Long Xiao. Dynamic Re-Sequencing of EV Platoons Using Noisy Dueling DQN for Energy Fairness. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 5248–5252, 2025. doi: 10.1109/SMC58881.2025.11342737.

Yongxin Zhu, Yongfu Li, Simon Hu, and Shuyou Yu. Optimal Control for Vehicle Platoon Considering External Disturbances. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 453–458, 2022. doi: 10.1109/ITSC55140.2022.9922486.

A Traffic Prediction and Classification Details

This appendix provides additional implementation details of the traffic-state prediction and classification module used by the safe-aware execution gate. The traffic condition at time t is represented by

$$\mathbf{x}_t = [F_t, V_t, K_t]^{\mathrm{T}}, \tag{19}$$

where F_t, V_t, and K_t denote traffic flow, average speed, and traffic density, respectively. The density satisfies

$$K_t = \frac{F_t}{V_t}. \tag{20}$$

Given an observation window of length L, the historical traffic sequence available at decision instant t_k is

$$\mathbf{X}_k = [\mathbf{x}_{t_k-L+1}, \mathbf{x}_{t_k-L+2}, \ldots, \mathbf{x}_{t_k}]. \tag{21}$$

An LSTM model is used to predict the traffic feature vector for the next decision stage:

$$\hat{\mathbf{x}}_{t_k+1} = f_{\mathrm{LSTM}}(\mathbf{X}_k; \theta_{\mathrm{LSTM}}), \tag{22}$$

where θ_{LSTM} denotes the trainable model parameters.

Given Q training sequences, the prediction network is trained by minimizing the mean squared prediction error,

$$\mathcal{L}_{\mathrm{LSTM}} = \frac{1}{Q}\sum_{q=1}^{Q} \|\mathbf{x}_{q+1} - \hat{\mathbf{x}}_{q+1}\|_2^2 . \tag{23}$$

Before training, each traffic feature is normalized using the statistics of the training set. The same normalization parameters are subsequently applied to the validation and test samples.

The predicted traffic vectors are categorized using fuzzy C-means (FCM). Let

$$U = [u_{ij}] \in [0,1]^{n\times g} \tag{24}$$

denote the membership matrix, where n is the number of samples and g is the number of traffic categories. Let

$$Z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_g] \tag{25}$$

denote the cluster centers. The FCM objective function is

$$J(U, Z) = \sum_{i=1}^{n}\sum_{j=1}^{g} u_{ij}^{m} \|\hat{\mathbf{x}}_i - \mathbf{z}_j\|_2^2 , \tag{26}$$

subject to

$$\sum_{j=1}^{g} u_{ij} = 1, \qquad 0 \le u_{ij} \le 1. \tag{27}$$

The membership degree is updated according to

$$u_{ij} = \left[\sum_{\ell=1}^{g} \left(\frac{\|\hat{\mathbf{x}}_i - \mathbf{z}_j\|_2}{\|\hat{\mathbf{x}}_i - \mathbf{z}_\ell\|_2}\right)^{\frac{2}{m-1}}\right]^{-1}, \tag{28}$$

and the cluster centers are updated by

$$\mathbf{z}_j = \frac{\sum_{i=1}^{n} u_{ij}^m \hat{\mathbf{x}}_i}{\sum_{i=1}^{n} u_{ij}^m}. \tag{29}$$

The iteration terminates when the variation of the cluster centers falls below a predefined tolerance. For an online predicted traffic vector, the discrete traffic category is determined by

$$\rho(t_k) = \arg \max_{1 \le j \le g} u_{kj}. \tag{30}$$

In this study, $g = 4$, corresponding to free flow, basic free flow, mild congestion, and severe congestion. Re-sequencing is considered traffic-feasible only when

$$\rho(t_k) \in \{1, 2\}, \qquad \ell(t_k) = 1, \tag{31}$$

where $\ell(t_k)$ denotes the lane-change permission indicator. Otherwise, the currently executed platoon order is maintained.

B ETM_1: Complete Triggering Rule and Inter-Event Analysis

Proof B.1 *Let $\{\tau_m\}_{m \in \mathbb{N}_{>0}}$ denote the sequence of triggering instants generated by ETM_1. Since the vehicle speeds and the position-dependent energy-saving coefficients are bounded, the SoC trajectories remain bounded over the admissible operating region. Consequently, the SoC-dispersion function $V(t)$ and the adaptive threshold $\Gamma(t)$ are locally Lipschitz. Hence, there exist finite constants $L_V > 0$ and $L_\Gamma > 0$ such that, for any t_1 and t_2 between two consecutive triggering events,*

$$|V(t_2) - V(t_1)| \le L_V |t_2 - t_1|, \tag{32}$$

and

$$|\Gamma(t_2) - \Gamma(t_1)| \le L_\Gamma |t_2 - t_1|. \tag{33}$$

Recall that the lower self-resetting threshold is defined as equation 8. Therefore, the gap between the triggering threshold and the self-resetting threshold is

$$\Delta(t) = \Gamma(t) - \underline{\Gamma}(t) = (1 - \rho_e)\Gamma(t). \tag{34}$$

From the definition of $\Gamma(t)$ in equation 6,

$$\Gamma(t) \ge \sigma^2, \tag{35}$$

and hence

$$\Delta(t) \ge (1 - \rho_e)\sigma^2 \triangleq \Delta_{\min} > 0. \tag{36}$$

To preserve the self-resetting property under persistent SoC imbalance while maintaining a strictly positive post-event triggering margin, define an auxiliary reset margin $\eta(t) \ge 0$. Immediately after the m-th triggering event, it is reset as

$$\eta(\tau_m^+) = V(\tau_m) - \Gamma(\tau_m) + \Delta(\tau_m). \tag{37}$$

After the reset, the margin is released according to

$$\eta(t) = \max\left\{0,\, \eta(\tau_m^+) - \kappa(t - \tau_m)\right\}, \kappa > 0, \tag{38}$$

until the next triggering event. The corresponding ETM_1 triggering condition is

$$V(t) \ge \Gamma(t) + \eta(t). \tag{39}$$

Define the triggering function

$$h_1(t) = V(t) - \Gamma(t) - \eta(t). \tag{40}$$

Immediately after the m-th event,

$$\begin{aligned} h_1(\tau_m^+) &= V(\tau_m) - \Gamma(\tau_m) - \eta(\tau_m^+) \\ &= -\Delta(\tau_m) \\ &\le -\Delta_{\min} < 0. \end{aligned} \tag{41}$$

Therefore, every triggering event generates a strictly positive separation from the next triggering surface.

Moreover, for any $t \geq \tau_m$ before the next event, the Lipschitz properties of $V(t)$ and $\Gamma(t)$ together with the release rate of $\eta(t)$ yield

$$h_1(t) - h_1(\tau_m^+) \leq (L_V + L_\Gamma + \kappa)(t - \tau_m). \tag{42}$$

Let

$$L_1 = L_V + L_\Gamma + \kappa. \tag{43}$$

At the next triggering instant τ_{m+1}, the triggering function must reach the triggering condition

$$h_1(\tau_{m+1}) \geq 0. \tag{44}$$

It therefore follows that

$$\Delta_{\min} \leq L_1(\tau_{m+1} - \tau_m), \tag{45}$$

which gives

$$\tau_{m+1} - \tau_m \geq \frac{\Delta_{\min}}{L_1} = \frac{(1-\rho_e)\sigma^2}{L_V + L_\Gamma + \kappa} > 0. \tag{46}$$

Hence, two consecutive ETM_1 triggering events are always separated by a strictly positive MIET. The event sequence therefore cannot admit a finite accumulation point, and infinitely many re-sequencing events cannot occur within any finite time interval. Thus, ETM_1 is Zeno-free.

C ETM_2: Cooldown Rule and Inter-Event Analysis

Proof C.1 *Let $\{\tau_m\}_{m\in\mathbb{N}_{>0}}$ denote the sequence of triggering instants generated by ETM_2. Suppose that the m-th re-sequencing event is generated at τ_m. At this instant, the imbalance-severity ratio is defined as equation 12. Since an event can be generated only when $V(\tau_m) \geq \Gamma(\tau_m)$ and $\Gamma(\tau_m) > 0$, it follows that*

$$0 < r_m \leq 1. \tag{47}$$

After the m-th triggering event, ETM_2 assigns an adaptive cooldown duration according to

$$T_m = \max\{T_{\min}, T_{\text{base}} r_m\}, \tag{48}$$

where $T_{\min} > 0$ denotes the minimum admissible cooldown duration and $T_{\text{base}} > 0$ is the base cooldown parameter. Therefore, for every triggering event,

$$T_m \geq T_{\min} > 0. \tag{49}$$

During the cooldown interval following τ_m, the ETM_2 gate remains locked and no new re-sequencing event can be generated. Consequently, the earliest admissible instant for the next event satisfies

$$\tau_{m+1} \geq \tau_m + T_m. \tag{50}$$

Thus,

$$\tau_{m+1} - \tau_m \geq T_m \geq T_{\min} > 0. \tag{51}$$

Hence, every pair of consecutive ETM_2 triggering events is separated by a strictly positive time interval. In particular, for any finite interval of length $T > 0$, the number of triggering events is bounded by

$$N_{\text{event}}(T) \leq 1 + \left\lfloor \frac{T}{T_{\min}} \right\rfloor, \tag{52}$$

which is finite. Therefore, the event sequence cannot have a finite accumulation point, and infinitely many re-sequencing events cannot occur within any finite time interval. Thus, ETM_2 is Zeno-free.

Table 1: ETM_1 parameters.

Parameter	Value
ϵ	5×10^{-3}
β	0.3
σ	1.5×10^{-3}
ρ_e	0.3

Table 2: ETM_2 parameters.

Parameter	Value
ϵ	5×10^{-6}
β	0.6
σ	1.5×10^{-6}
ρ_e	0.3
$C_{\min}$	2
C_{base}	15

D DQN Training and Algorithmic Details

At decision instant t_k, the DQN state is defined as

$$\boldsymbol{\zeta}_k = [\mathbf{s}(t_k), \mathbf{P}(k), d_k^{\text{norm}}], \tag{53}$$

where $\mathbf{s}(t_k)$ represents the current SoC distribution, $\mathbf{P}(k)$ is the currently executed platoon order, and d_k^{norm} represents the normalized travel progress. For a platoon containing N EVs, each feasible permutation is treated as one discrete action:

$$\mathcal{A} \equiv \mathcal{P}, \qquad |\mathcal{A}| = N!. \tag{54}$$

The greedy candidate platoon order is

$$\widetilde{\mathbf{P}}(k) = \arg\max_{\mathbf{P}' \in \mathcal{P}} Q(\boldsymbol{\zeta}_k, \mathbf{P}'; \theta). \tag{55}$$

During training, an ϵ-greedy policy is used:

$$\widetilde{\mathbf{P}}(k) = \begin{cases} \mathbf{P}_{\text{rand}}, & \xi_k < \epsilon_k, \\ \arg\max_{\mathbf{P}' \in \mathcal{P}} Q(\boldsymbol{\zeta}_k, \mathbf{P}'; \theta), & \xi_k \geq \epsilon_k, \end{cases} \tag{56}$$

where $\xi_k \sim \mathcal{U}(0,1)$. Each interaction generates an experience tuple

$$\mathcal{E}_k = (\boldsymbol{\zeta}_k, \widetilde{\mathbf{P}}(k), r_k, \boldsymbol{\zeta}_{k+1}, \chi_k), \tag{57}$$

where $\chi_k \in \{0,1\}$ is the terminal indicator. The temporal-difference target is

$$y_j = r_j + \gamma(1-\chi_j) \max_{\mathbf{P}' \in \mathcal{P}} Q(\boldsymbol{\zeta}_{j+1}, \mathbf{P}'; \theta^-), \tag{58}$$

and the Q-network is optimized by minimizing

$$\mathcal{L}_{\text{DQN}} = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \Big[y_j - Q(\boldsymbol{\zeta}_j, \widetilde{\mathbf{P}}(j); \theta) \Big]^2. \tag{59}$$

The online-network parameters are updated using

$$\theta \leftarrow \theta - \alpha_Q \nabla_\theta \mathcal{L}_{\text{DQN}}, \tag{60}$$

and the target parameters are periodically synchronized as

$$\theta^- \leftarrow \theta. \tag{61}$$

Table 3: DQN training parameters.

Parameter	Value
Number of hidden layers	2
Hidden units per layer	128
Discount factor γ	0.99
Exploration ϵ_0	0.3
Exploration decay	0.98
Target update interval	10
Training episodes	12,000
Replay-buffer size	10,000
Batch size	64

E Extended Framework Illustration

Fig. 1 illustrates the execution procedure of the proposed DQN-based safe-aware platoon re-sequencing algorithm. According to Algorithm 1, the current platoon state is first used to generate a candidate platoon order through the DQN policy. The obtained candidate order $\widetilde{P}(t_k)$ is then evaluated by the safety-aware execution module. If the predicted traffic condition, lane permission, and ETM are simultaneously satisfied, the candidate order is applied; otherwise, the current platoon order is retained.

As shown in the Fig. 1, the re-sequencing process changes the vehicle-position mapping from the original platoon configuration $P(k) = [1, 2, 3, 4]$ to the executed order $P(k+1) = [2, 1, 3, 4]$. Accordingly, the position index of vehicle 1 changes from $P(1, k) = 1$ to $P(1, k+1) = 2$, while vehicle 2 changes from $P(2, k) = 2$ to $P(2, k+1) = 1$. The other vehicles maintain their original positions. This position update changes the aerodynamic environment experienced by each vehicle and consequently modifies the corresponding energy consumption rate.

The aerodynamic effect is also illustrated in Fig. 1. The red arrows in front of the leading vehicle represent a higher aerodynamic drag region because the leading vehicle directly encounters the incoming airflow. In contrast, the blue arrows behind preceding vehicles indicate a lower drag region caused by the wake effect. Therefore, different platoon positions provide different aerodynamic benefits, and the variation of $P(i, k)$ determines the allocation of these benefits among vehicles. Through the above re-sequencing mechanism, the proposed algorithm dynamically adjusts vehicle positions according to the learned candidate order while ensuring safe execution, thereby improving energy efficiency and balancing the terminal SoC distribution.

F Simulation Configuration and Reproducibility

All compared methods are evaluated using the same vehicle parameters, traffic scenarios, and termination conditions. Each EV is initialized with normalized SoC

$$s_i(t_0) = 1. \tag{62}$$

The battery capacity is

$$\Lambda = 85 \text{ kWh}, \tag{63}$$

and the baseline energy-consumption rate is

$$\delta = 211 \text{ Wh/km}. \tag{64}$$

The position-dependent aerodynamic energy-saving coefficients are

$$e_1 = 0.043, \quad e_2 = 0.10, \quad e_3 = 0.14, \quad e_4 = 0.14. \tag{65}$$

For platoons containing more than four EVs, positions beyond the fourth position are assigned

$$e_i = 0.14, \qquad i \geq 4. \tag{66}$$

The traffic prediction module is trained and evaluated using data from the California Performance Measurement System (PeMS). The dataset provides traffic flow and speed measurements collected from freeway detectors, from which traffic density is derived according to $K = F/V$.

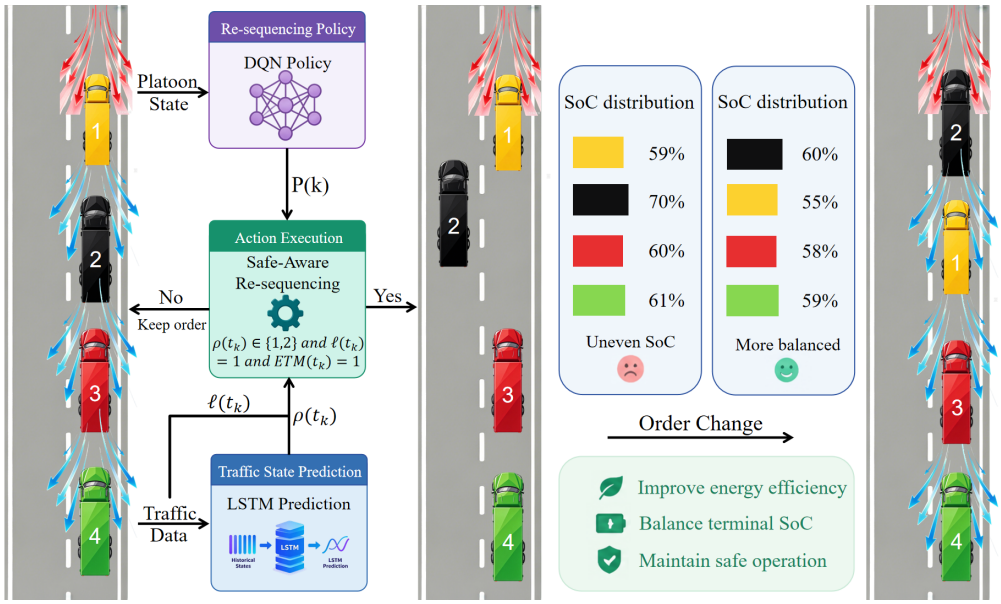

Figure 1: Illustration of the proposed DQN-based safe-aware platoon re-sequencing framework. The red and blue arrows denote high and low aerodynamic drag regions, respectively, corresponding to the exposed airflow at the leading position and the wake effect of preceding vehicles.

Algorithm 1 DQN-Based Safe-Aware Platoon Re-sequencing Algorithm

Require: Vehicle set Ω, feasible platoon orders $\mathcal{P}$, DQN parameters θ, safety constraints
Ensure: Executed platoon sequence $\{\mathbf{P}(k)\}$
Initialize Q-network $Q(\cdot,\cdot;\theta)$ and replay buffer $\mathcal{D}$
Initialize platoon state $\mathbf{s}(t_0)$ and order $\mathbf{P}(0)$
for each training episode **do**
  for each decision instant t_k **do**
    Construct state $\boldsymbol{\zeta}_k$
    Select candidate order $\widetilde{\mathbf{P}}(k)$ using ϵ-greedy policy
    Execute candidate order in simulation environment
    Calculate reward r_k based on SoC balancing objective
    Store transition $(\boldsymbol{\zeta}_k, \widetilde{\mathbf{P}}(k), r_k, \boldsymbol{\zeta}_{k+1})$ into $\mathcal{D}$
    Update DQN parameters using sampled mini-batch
  end for
end for
At online operation, generate candidate order:

$$\widetilde{\mathbf{P}}(k) = \arg\max_{\mathbf{P}' \in \mathcal{P}} Q(\boldsymbol{\zeta}_k, \mathbf{P}'; \theta)$$

Obtain traffic state $\rho(t_k)$, lane permission $\ell(t_k)$, and triggering condition $ETM(t_k)$
if $\rho(t_k) \in \{1,2\}$, $\ell(t_k) = 1$, and $ETM(t_k) = 1$ **then**
  $\mathbf{P}(k) = \widetilde{\mathbf{P}}(k)$
else
  $\mathbf{P}(k) = \mathbf{P}(k-1)$
end if

The proposed DQN candidate-order policy is compared with Monte Carlo, Q-learning, TRPO, and SOC-Ranking. All methods are evaluated under the same vehicle model and traffic scenarios. When ETM_1 or ETM_2 is used, the same event-triggering rule is applied to all candidate-order policies unless otherwise stated.

Table 4: Vehicle and simulation parameters.

Parameter	Value
Platoon size N	$4, 5, 6, 7$
Initial SoC	1.0
Minimum SoC $s_{\min}$	0
Battery capacity Λ	85 kWh
Baseline energy rate δ	211 Wh/km
e_1	0.043
e_2	0.10
e_3	0.14
$e_i,\ i \geq 4$	0.14

Table 5: Performance comparison of different re-sequencing algorithms under ETM_1 with different platoon sizes

Algorithm	Episode Reward				Total Travel Distance (km)				Number of Re-sequencing Operations			
	4 EVs	5 EVs	6 EVs	7 EVs	4 EVs	5 EVs	6 EVs	7 EVs	4 EVs	5 EVs	6 EVs	7 EVs
DQN	**-6.47**	**-5.42**	**-6.79**	**-6.30**	**270.85**	**276.63**	**267.96**	**269.04**	**3**	**3**	**3**	**4**
Monte Carlo Guo et al. (2022)	-7.16	-7.11	-8.17	-6.55	273.63	269.17	266.67	269.17	6	6	6	5
Q-learning	-5.02	-8.87	-8.55	-8.39	274.56	266.67	267.76	266.67	5	4	3	3
TRPO Schulman et al. (2017)	-7.45	-4.99	-6.63	-7.12	269.04	273.63	269.17	267.96	3	3	6	5
SOCRANKING Guo et al. (2022)	-6.32	-7.32	-6.98	-7.83	269.12	275.54	270.62	268.71	3	5	3	3

Table 6: Performance comparison of different re-sequencing algorithms under ETM_2 with different platoon sizes

Algorithm	Episode Reward				Total Travel Distance (km)				Number of Re-sequencing Operations			
	4 EVs	5 EVs	6 EVs	7 EVs	4 EVs	5 EVs	6 EVs	7 EVs	4 EVs	5 EVs	6 EVs	7 EVs
DQN	**-3.30**	**-1.14**	**-2.79**	**-1.72**	**274.24**	**280.74**	**275.38**	**283.95**	**5**	**5**	**4**	**4**
Monte Carlo	-3.58	-1.56	-1.73	-1.85	274.56	278.55	278.55	275.93	6	6	7	5
Q-learning	-1.73	-1.27	-2.09	-3.01	278.55	279.81	277.55	274.23	5	7	6	5
TRPO	-3.31	-1.53	-1.93	-2.67	273.63	275.93	278.54	273.61	4	6	6	5
SOCRANKING	-2.74	-1.32	-2.92	-2.08	278.55	279.81	275.02	278.52	4	4	6	4

G Additional Experimental Results

Across all tested platoon sizes, the DQN policy obtains substantially higher episode rewards under ETM_2 than under ETM_1. Specifically, its reward changes from $(-6.47, -5.42, -6.79, -6.30)$ under ETM_1 to $(-3.30, -1.14, -2.79, -1.72)$ under ETM_2 for platoon sizes from four to seven vehicles. This improvement is accompanied by an increase in the number of executed re-sequencing operations. For DQN, the operation count changes from $(3, 3, 3, 4)$ under ETM_1 to $(5, 5, 4, 4)$ under ETM_2. Therefore, the advantage of ETM_2 should be interpreted as a more effective reward-execution trade-off rather than a reduction in the absolute number of re-sequencing events.

No single candidate-order policy dominates all platoon sizes in terms of episode reward. Nevertheless, DQN exhibits a favorable overall trade-off between reward performance and the frequency of re-sequencing operations, while its discrete action formulation is naturally compatible with the finite platoon-permutation space. For this reason, DQN is adopted as the candidate-order generator in the proposed framework.

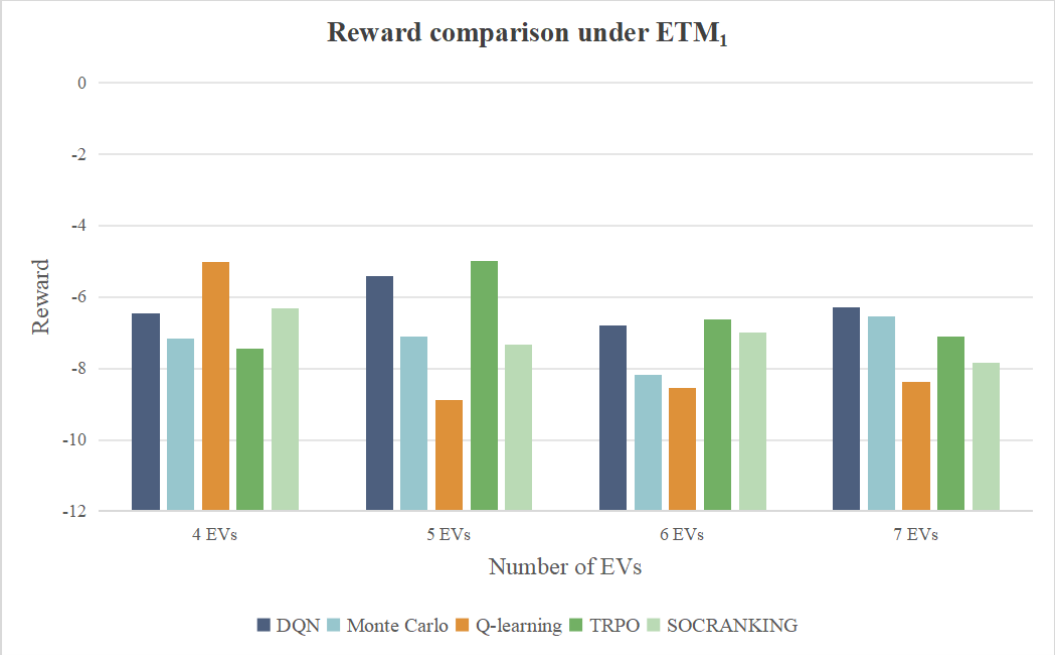

Figure 2: Episode reward comparison of different re-sequencing algorithms under ETM_1 with different platoon sizes.

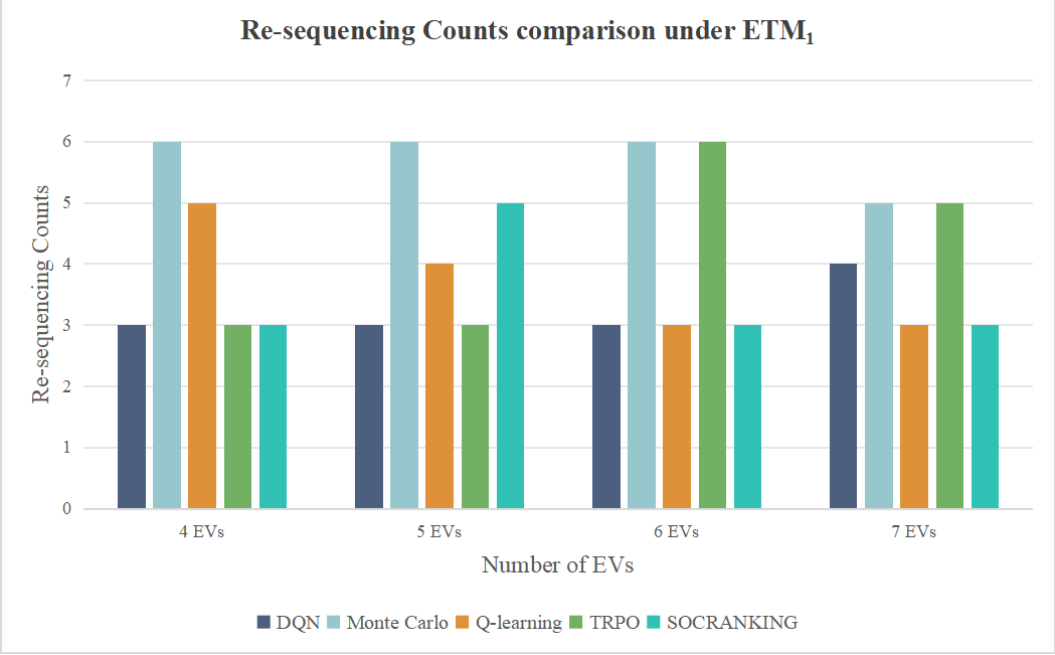

Figure 3: Number of re-sequencing operations of different algorithms under ETM_1 with different platoon sizes.

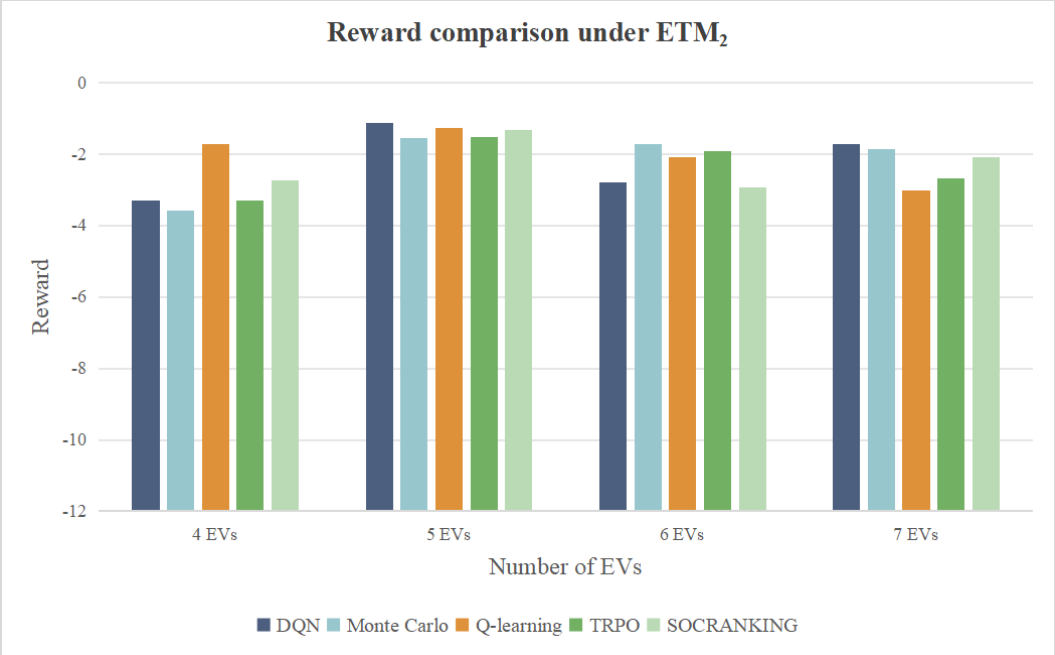

Figure 4: Episode reward comparison of different re-sequencing algorithms under ETM_2 with different platoon sizes.

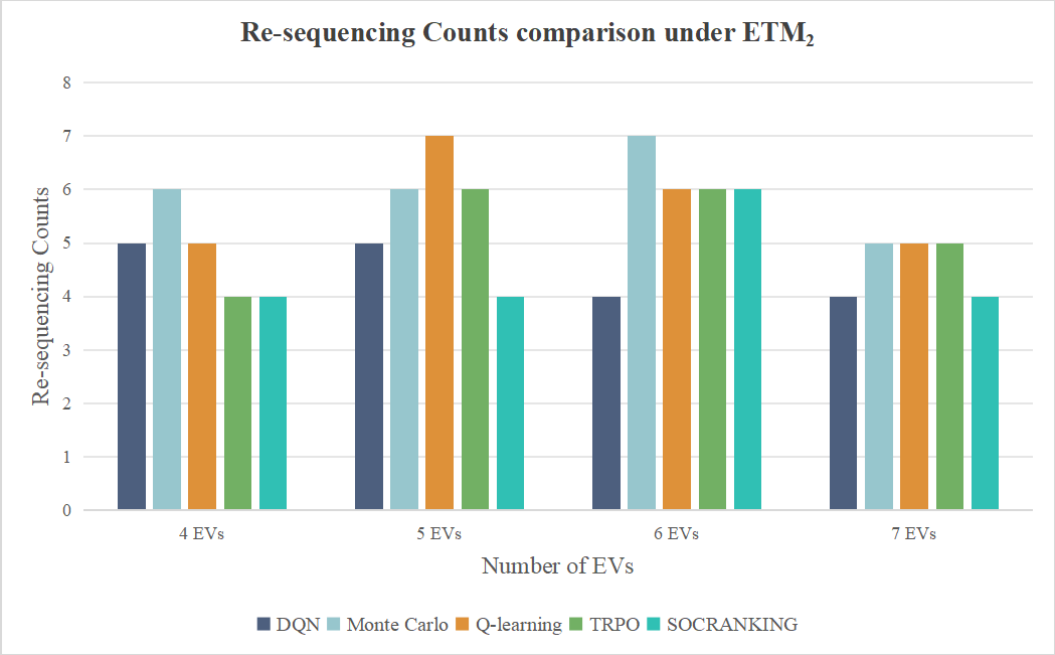

Figure 5: Number of re-sequencing operations of different algorithms under ETM_2 with different platoon sizes.