Wiley
Journal of Advanced Transportation
Volume 2025, Article ID 2838943, 14 pages
https://doi.org/10.1155/atr/2838943

WILEY

Research Article

Reinforcement Learning–Based Ramp Metering Strategy Considering Queue Management

Yang Yang,[1,2] **Shixuan Yu**,[3] **Fan Ding**,[3] **and Yu Han**[3]

[1]*Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai, China*
[2]*Intelligent Transportation Research Center, Jiangsu SINOROAD Engineering Research Institute Co., Ltd., Nanjing, China*
[3]*School of Transportation, Southeast University, Nanjing, China*

Correspondence should be addressed to Yu Han; yuhan@seu.edu.cn

Received 16 January 2024; Revised 25 December 2024; Accepted 26 December 2024

Academic Editor: Tomio Miwa

This paper introduces an action replacement module for reinforcement learning (RL)–based ramp metering to address the issue of ramp queue spillback during the training process. Ramp queue spillback leads to significant impacts on the traffic efficiency of adjacent road networks, making it a critical concern in ramp control. Existing RL approaches often employ ramp states as reward functions to encourage agents to learn strategies that avoid queue overflow. However, due to the trial-and-error nature of RL, these methods frequently generate actions that cause queue spillback during training, posing challenges for real-time online training in real-world applications. To overcome this limitation, the proposed action replacement module utilizes the store-and-forward model to estimate a lower bound for ramp metering rates. By identifying and replacing actions that fail to meet this constraint, the strategy effectively prevents queue spillback. In addition, penalties are imposed on replaced actions to guide the agent in learning effective and practical control policies. The proposed method is evaluated in both single-ramp and multiramp scenarios. Experimental results demonstrate that the agent can learn the queue spillback prevention strategies, and nearly eliminate ramp queue spillback without compromising control performance.

1. Introduction

Ramp metering is a commonly adopted traffic control strategy on freeways and has been successfully applied in many countries [1, 2]. This control strategy regulates the merging flow from on-ramps to the mainline through the ramp traffic signals, aiming to mitigate congestion and improve traffic efficiency.

Feedback control strategies are widely utilized in ramp metering due to their simplicity and reliability. A notable example is Asservissement Linéaire d'Entrée Autoroutière (ALINEA) [3]. ALINEA incorporates a feedback loop that adjusts the current's downstream occupancy based on the deviation from the critical occupancy. Due to its ease of implementation, numerous variants of ALINEA have been proposed to address traffic management under various scenarios: Wang et al. [4] proposed PI-ALINEA to tackle the distant downstream bottleneck problem and Smaragdis and Papageorgiou [5] proposed a flow-based version, FL-ALINEA; upstream-based version, UP-ALINEA; and queue management version, X-ALINEA/Q. Frejo and De Schutter [6] enhanced ALINEA by introducing a feed-forward structure, enabling control actions based on the predicted density of the bottleneck. To further improve global optimization and address congestion over larger areas, the heuristic ramp–metering coordination (HERO) strategy [7] was developed, which coordinates metering rates for both upstream and downstream ramps to balance queue lengths across multiple ramps. Despite their successes, feedback-based control strategies are inherently limited by their single-step decision-making structure and struggle to capture the rapid changes in traffic flow dynamics.

In contrast, model predictive control (MPC) strategies are designed to overcome the shortcomings by

incorporating predictions of future traffic states into the decision-making process. For example, Hegyi, De Schutter, and Hellendoorn [8] combined ramp metering and variable speed limits' control and developed a model predictive controller using METANET as the predictive model to minimize the total time spent (TTS) while minimizing variations in metering rate and speed limits. Similarly, Han et al. [9] utilized the macroscopic fundamental diagram model, extended cell transmission model, and METANET as predictive models to construct a hierarchical ramp coordination metering strategy based on MPC. Heshami and Kattan [10] modeled the ramp metering problem as a stochastic distributed MPC approach, solving it using a bargaining game framework. While MPC offers significant improvements over feedback-based methods, it requires substantial computational resources. Furthermore, there may exist a substantial mismatch between the physical model upon which the predictor relies and the actual environmental conditions, potentially resulting in suboptimal control performance.

Reinforcement learning (RL)–based traffic control strategies have gained widespread attention due to their ability to directly learn the policies from collected data, avoiding the mismatch between the predictive models and the actual environments often encountered with MPC [11]. These RL–based strategies offer significant advantages in dynamic traffic environments where conventional approaches might struggle. For instance, Belletti et al. [12] integrated Lighthill–Whitham–Richards partial differential equations into a multiagent RL–based ramp metering strategy, achieving similar control performance to the well-established ALINEA. Liu et al. [13] utilized traffic video data as high dimensional input to let the agent learn the policy in an end-to-end fashion. Han et al. [14] proposed a physics-informed RL–based ramp metering strategy to reduce the training cost of RL. Wang et al. [15] proposed a centralized RL framework to simultaneously optimize ramp metering and variable speed limits. Despite the promising results of RL–based control strategies, they often fail to address the ramp queue spillback, a frequently occurring phenomenon that not only causes excessive delays for the queued vehicles but also negatively impacts the traffic efficiency of adjacent road networks. A common approach to this challenge is incorporating ramp queue states into the reward functions. For instance, Davarynejad et al. [16] designed a reward function that accounts for the queue length limitations using the Q-learning algorithm and conducted simulation experiments with macroscopic traffic simulation. Lu and Huang [17] scaled the reward through maximum queue length to apply the penalties when the queue length exceeds the predefined ramp constraint. Deng et al. [18] introduced a queue spillback penalty term attached to the reward function to regulate policy behaviors. Cheng et al. [19] designed an overflow protection module to adaptively address ramp overflow issues. Table 1 provides a comprehensive summary of the various methods in ramp metering.

Although existing RL–based strategies incorporate various forms of queue spillback penalties into the reward functions, they are insufficient to prevent queue spillback throughout the entire training process due to the trial-and-error nature of RL. Inappropriate actions are still taken by RL agents, especially during the exploration phase in the early stages of training. This limitation hinders the applicability of such RL–based strategies for online training in real-world scenarios. A series of safe RL algorithms have been developed to regulate the agent's actions and decrease the safety risk during the policy exploration through modifying the network structure [20, 21] or adjusting the training procedure [22, 23]. However, there is little literature applying these methods in RL–based ramp metering strategies to prevent the ramp queue spillback.

Inspired by safe RL, this paper integrates an action replacement module into the RL–based ramp metering to prevent ramp queue spillback during the training process. Specifically, the proposed module comprises two key components: a lower bound constraint on-ramp metering rates and an action replacement policy. The first component utilizes a store-forward model to estimate the minimum merging rate that avoids ramp queue spillback, while the second component employs this lower bound to identify the risky actions taken by the RL agent. These actions are then replaced and the associated environment rewards are adjusted according to the replacement policy. The proposed module is evaluated in both single-ramp and multiramp scenarios, using microscopic simulations of real-world road networks. Comparative experiments are conducted against several state-of-the-art baselines including ALINEA, HERO, and RL–based control with a queue spillback penalty term. The results demonstrate that the proposed method can effectively eliminate ramp queue spillback during the training process without compromising control performance and can successfully guide the agent to learn the policy that avoids spillback.

2. Problem Statement

2.1. Capacity Drop Phenomenon. The basic idea of ramp metering is to regulate the ramp flow entering the mainline so that the outflow downstream of the merge bottleneck matches the road capacity. The reason the merging flow needs to be carefully adjusted is that if the combined flow of the mainline and the ramp exceeds the road capacity, it can cause a sudden drop in downstream capacity. Research has indicated that the freeway traffic flow is discontinuous near the capacity point on the flow-density diagram, displaying an inverse-λ form. After the critical density is exceeded, the upstream of the bottleneck is set in a congested state and the downstream of the bottleneck suffers a sudden drop in discharge flow. The capacity drop phenomenon hinders us from fully utilizing the traffic infrastructure to discharge the traffic flow, thus further exacerbating congestion at the bottleneck. Researchers have analyzed freeway traffic flow data from various countries and found that capacity drop typically ranges from 2% to 30% for the bottlenecks [24–26]. The cumulative arrival curves calculated by the following equation are employed to analyze the capacity drop [27, 28]:

$$N'(x,t) = N(x,t) - q_0 \times t, \tag{1}$$

Table 1: The summary of the literature review on RL–based ramp metering.

Literature	Method	Control scenario	Queue management
[12]	Reinforce	Coordinated ramp metering	No queue management
[13]	DQN	Ramp metering	No queue management
[15]	DDPG/TD3	Coordinated ramp metering and variable speed limit	No queue management
[14]	Q-learning/DQN/BCQ	Coordinated ramp metering	No queue management
[16]	Q-learning	Ramp metering	Scale reward function by queue length
[18]	MAPPO	Coordinated ramp metering	Add penalty term to reward function
[17]	Q-learning	Ramp metering	Scale reward function by max queue length
[19]	DQN	Coordinated ramp metering	Adjust the signal control directly

where N and N' represent the cumulative number of vehicles that arrived before and after modification subtracting the background flow, while q_0 denotes the background flow.

2.2. Ramp Metering Problem. The primary objective of ramp metering is to minimize the TTS by vehicles on the network within a specified time interval. The following equations can be formed:

$$\min_{r} \text{TTS} = T \times \sum_{k=0}^{K} N(k),$$
$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{r}(k), \boldsymbol{d}(k)], \\ 0 \le \text{queue}_i \le \text{queue}_{i,\max}, \\ r_{i,\min} \le r_i(k) \le r_{i,\max}, \end{cases} \tag{2}$$

where T represents the control step length, $N(\cdot)$ represents the vehicle number of the network, and $\boldsymbol{x}(k), \boldsymbol{r}(k), \boldsymbol{d}(k)$ represent the traffic state vector, ramp metering rate vector, traffic demand vector (both mainline and on-ramp), respectively. The first constrain indicates that the next step traffic state can be inferred by the state, ramp metering rate, and traffic demand in the current step; the second constrain shows that the queue length of on-ramps is limited; and the third constrain shows that the ramp metering rates are bounded.

3. Methodology

3.1. Feedback Control Strategy. The feedback traffic control strategies have proved effective in many studies. Here, we introduce the most frequently used feedback strategy ALINEA and its multiramp coordinated version HERO.

ALINEA calculates the ramp metering rate by utilizing the discrepancy between the actual occupancy and the critical occupancy as a feedback signal for the next control step. By adjusting the metering rate, ALINEA can maintain the downstream occupancy near the critical value. The control formula of ALINEA is as follows:

$$r(k) = r(k-1) + k_r \times \left[\widehat{O} - O_{\text{out}}(k)\right], \tag{3}$$

where k_r is a positive regulator parameter, $\widehat{O}$ is critical occupancy, and O_{out} is actual occupancy downstream of the merging area.

In the HERO system, each ramp determines whether to activate coordinated control based on the corresponding queue length and the downstream occupancy. Coordinated control is triggered when the queue length or downstream occupancy exceeds a threshold. Through reducing the metering rate of the upstream ramp involved in the coordinated control, the traffic flow on the mainline is reduced, thus allowing vehicles of the downstream ramp to merge smoothly. When the queue length and downstream occupancy decrease, the coordinated control deactivates, and the HERO system degrades into multiple ALINEA systems. At this point, the queue length percentage of each ramp is approximately balanced.

3.2. RL Algorithm. RL involves a continuous interaction between the agent and the environment, aiming to find a policy that maximizes the cumulative reward obtained from the environment. The Markov decision process (MDP) is used to describe such interaction. The components of MDP can be represented by a tuple: $M = <S, A, P, R, \gamma>$. S represents a general description of the current environment and A represents action, namely, the decision made by the agent based on the state and the learned policy. The decision-making process of an agent can be regarded as a mapping from state to control action, denoted as $\pi(a \mid s)$: $S \longrightarrow A$. P is a transition probability matrix, indicating the probability from the current state s to the next state s' after taking an action a, denoted as $P(S' = s' \mid S = s, A = a)$. In the context of the MDP, the variables exhibit the Markov property, where s_{t+1} is only dependent on s_t and independent on s_i $(i < t)$. R represents the reward, which serves as a metric to assess the agent's decision-making. Higher rewards suggest closer proximity to the goal, while lower values suggest deviation from it. Due to the varying importance of immediate and future rewards, we apply a discount to the anticipated future reward when calculating the cumulative reward. The specific formula of cumulative reward is as follows:

$$G(t) = R_t + \gamma \times R_{t+1} + \gamma^2 \times R_{t+2} + \cdots + \gamma^{N-t} \times R_N = \sum_{i=t}^{N} \gamma^{i-t} R_i, \tag{4}$$

where γ is the discount to the future reward, which is the last component of MDP. We can utilize the cumulative reward to construct an action-value function that evaluates the quality of choosing a particular action in a given state as

$$Q_\pi(s_t, a_t) = E\left[G(t) \mid S_t = s_t, A_t = a_t\right]. \tag{5}$$

Similarly, we can also utilize the action-value function to construct a value function to evaluate the current state as follows:

$$V_\pi(s_t) = E_{A_t \sim \pi(\cdot \mid s_t)}\left[Q_\pi(s_t, A_t) \mid S_t = s_t\right]. \tag{6}$$

Policy-based learning is one of the most commonly used frameworks in RL. These kinds of algorithms use a neural network $\pi(a \mid s; \theta)$ to learn the agent policy, and according to the policy gradient theorem [29], the following function can be used to update the neural network parameters θ:

$$\nabla J(\theta) = E_S\left\{E_{A \sim \pi(\cdot \mid s;\theta)}\left[Q_\pi(S, A) \nabla_\theta \ln \pi(A|S;\theta)\right]\right\}. \tag{7}$$

The origin policy gradient algorithm uses Monte Carlo approximation to compute the action values and its stochastic gradient by collecting trajectory data through the interaction between the agent and the environment. However, this algorithm suffers from low sample efficiency and introduces significant variance during sampling, making it difficult to converge during training. To address these issues, the trust region policy gradient (TRPO) method [30] modifies equation (7) by introducing importance sampling to enhance data utilization. It replaces the action-value function with the advantage function A_t to reduce the variance. The advantage function A_t is defined as the difference between the action-value function Q_t and the value function V_t. Furthermore, TRPO incorporates a KL divergence penalty in the target function to prevent significant differences in the policy network before and after parameter updates. The optimization objective is as follows:

$$\max_\theta \widehat{E}_t\left\{\frac{\pi\left(a_t|s_t;\theta\right)}{\pi\left(a_t|s_t;\theta_{\text{old}}\right)} \times \widehat{A}_t - \beta \text{KL}\left[\pi(\cdot|s_t;\theta_{\text{old}}) \mid \pi(\cdot|s_t;\theta)\right]\right\}, \tag{8}$$

where β is the penalty factor. The implementation of TRPO is relatively complex, and the intricate constraints on policy updates (KL divergence in equation (8)) along with the optimization steps result in a considerable computational load. As shown in equation (9), the proximal policy optimization (PPO) [31] no longer computes the KL divergence directly, it utilizes a clipped importance sampling ratio $r_t(\theta)$ to prevent significant differences during policy network updates.

$$L(\theta) = \widehat{E}\left\{\min\left[r_t(\theta) \times \widehat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\widehat{A}_t\right]\right\}, \tag{9}$$

where ϵ is a hyperparameter. Therefore, PPO has the advantages of simplicity, efficiency, and robustness, making it a widely used RL algorithm.

3.3. RL–Based Control Strategy. In this section, we proposed an RL–based ramp metering strategy using the PPO algorithm. Figure 1 shows the structure of the neural networks that are used in the experiments; the first few fully connected layers (FC layers) of the actor network and the critic network share the parameters. After the shared network, each has its independent neural network to produce different outputs. Hyperbolic tangent (Tanh) is the activation function between each FC layer. The advantage of using a shared network is that it allows the actor and critic networks to share the same feature extractor, which helps improve the network's representation ability for observation data and facilitates information integration between the two networks.

3.3.1. Environment State. Flow, speed, and density are the three important parameters for describing traffic conditions on a freeway. However, loop detectors used in traffic monitoring systems cannot directly provide density measurements for specific road segments, but they can provide occupancy for the given cross-section. Assuming all vehicles have the same length, the occupancy is directly proportional to density: if N vehicles pass the detector within a detection cycle T, and the speed of the i th vehicle is denoted as v_i, we have

$$O = (d+l) \times \frac{N}{T} \times \frac{1}{N}\sum_{i=1}^{N}\frac{1}{v_i} = (d+l) \times \frac{q}{\overline{v}_s} = (d+l) \times k, \tag{10}$$

where d and l are the length of the detector and vehicle, respectively; $\overline{v}_s$ is the space mean speed; and k is the density. The environment state consists of two aspects: mainline state and ramp state. The mainline state includes the occupancy, speed, and number of passing vehicles detected by all cross-sectional loop detectors upstream and downstream of the merging area. The ramp state includes the queue length, the number of arrival and departure vehicles, and the green duration of the traffic lights in the last control step. During the experiment, we perform standardization on the states by subtracting the mean and dividing by the standard deviation before feeding them into the neural network.

3.3.2. Agent Action. The action of the agent is defined as the ratio of the ramp metering rate to the ramp capacity, which means that the action is a continuous value range in $[0, 1]$. Assuming that the discharge flow of the queue on the ramp equals capacity, the agent action can be interpreted as the percentage of the green light duration of the traffic signal. In our experimental setup, we sample the action to be executed from the normal distribution with the actor-network output as the expectation.

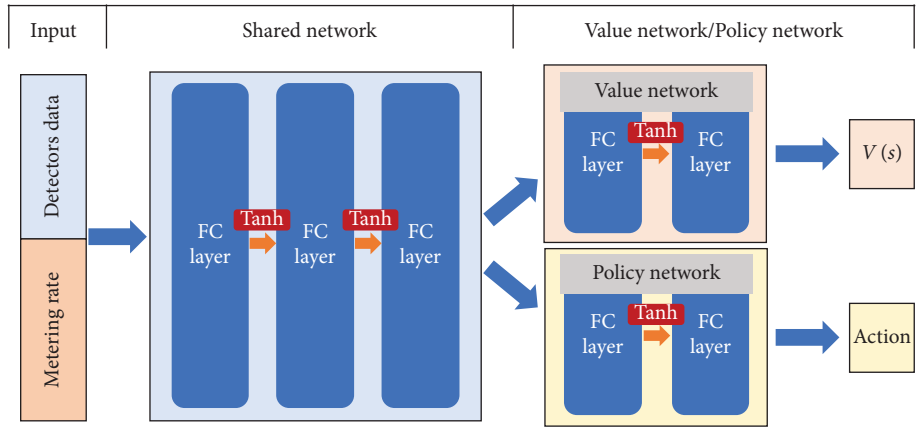

Figure 1: Structure of the neural network.

3.3.3. Environment Reward. The optimization goal of a RL agent is to maximize the cumulative reward. In traffic operation systems, objectives typically emphasize efficiency, safety, and sustainability [32]. In this study, we focus solely on traffic efficiency. Consequently, the proposed RL–based ramp metering strategy is designed to minimize the TTS. To achieve this, we define the reward function as a function of TTS as follows:

$$\text{reward}(k) = \frac{\alpha - \text{TTS}(k)}{\beta}, \tag{11}$$

where TTS can be obtained by summing up the number of vehicles on the road network for each simulation step k. α and β are the two constants. In the experiment, we replace the constants α and β with the maximum TTS and average TTS for a single control step under the baseline strategy, respectively.

3.4. Action Replacement Module With the Metering Rate Lower Bound Estimation. In freeway systems, the storage capacity of on-ramps is inherently limited. Once the number of queue vehicles exceeds the on-ramp storage capacity, spillback occurs, causing significant disruptions to adjacent networks. Therefore, minimizing the likelihood of actions leading to spillback is crucial when designing RL–based ramp metering strategies. In this section, we introduce the action replacement module that integrates ramp metering rate constraints into the RL training process to reduce the risk of ramp queue spillback caused by inappropriate agent actions. To better understand this approach, actions can be categorized into two types: safe actions, which do not cause spillback or drive the ramp traffic state toward spillback, and unsafe actions, which either directly result in spillback or push the traffic state closer to spillback. When the agent performs an unsafe action, it is essential to replace it with a safe alternative based on a well-defined strategy. The replaced action is then used to interact with the environment, preventing spillback. To achieve this functionality, we introduce two key modules: a lower bound constraint of ramp metering rate and an action replacement module. It is important to note that replacing only the action that immediately causes the spillback is insufficient because the ramp queue spillback is often the cumulative result of a sequence of unsafe actions, rather than a single misstep. Hence, a lower bound for the ramp metering rate is required to identify and replace unsafe actions consistently.

To compute the lower bound for the ramp metering rate, we utilize the store-and-forward model to get the most conservative rate as follows:

$$r_q(k) \geq \max\left[r_{\min}, d(k-1) - \frac{1}{T} \times (\widehat{w} - w(k))\right], \tag{12}$$

where $\widehat{w}$ is the allowed maximum queue length and $w(k)$ is the queue length in the k th control step. $(1/T) \times (\widehat{w} - w(k)$ calculates the maximum flow that can be accommodated by the on-ramp, so the above equation provides the critical ramp metering rate that happens to avoid spillback. Setting $\widehat{w} = w_{\max}$ ($w_{\max}$ represents the theoretical maximum number of vehicles a ramp can store, calculated as the ratio of the ramp length to the sum of vehicle length and the minimum gap between vehicles), this equation can be considered as providing a lower bound constraint for the ramp metering rate. However, due to variations in vehicle spacing, the actual queue length may fall below $w_{\max}$. To account for this, $\widehat{w}$ is adjusted by introducing an adjustment coefficient $\alpha < 1$ as follows:

$$r_q(k) \geq \max\left[r_{\min}, d(k-1) - \frac{1}{T} \times (\alpha \times w_{\max} - w(k))\right]. \tag{13}$$

The adjustment coefficient ensures a more realistic estimate of the lower bound. When there is no queue on the ramps, the second term in $\min(\cdot)$ may be smaller than 0, while $r_{\min}$ can prevent the complete close of on-ramps by providing the minimum merging flow. If the metering rate provided by the agent is smaller than the lower bound specified in equation (13), we regard such action as unsafe action. It is worth emphasizing that such unsafe action does not necessarily result in immediate spillback, but it contributes to traffic states that progressively lead to spillback over subsequent steps.

The action replacement module replaces each unsafe metering rate r with the lower bound metering rate r_{lb} and uses r_{lb} to interact with the environment. The RL process with the lower bound constraint of ramp metering rate is illustrated in Figure 2. To ensure the agent actively learns to avoid unsafe actions instead of passively relying on action replacement, a reward replacement module is incorporated. This module penalizes the agent based on the proximity to the unsafe region. The penalty is calculated using the predicted number of steps before spillback would occur if the agent's current action $r(k)$ and traffic demand $d(k)$ persist, i.e.,

$$\mathrm{penalty}(k) = \frac{w(k)}{k_{\mathrm{sp}} - k + 1}, \tag{14}$$

where k_{sp} is the step at which spillback is predicted to occur, determined iteratively by the queue length dynamics $w(k+1) = w(k) + T \times (d(k) - r(k))$. The overall reward function is then defined as follows:

$$R = \begin{cases} \mathrm{reward}(k), & \mathrm{action} \geq r_{\mathrm{lb}}, \\ \mathrm{reward}(k) - c \times \mathrm{penalty}(k), & \mathrm{action} < r_{\mathrm{lb}}, \end{cases} \tag{15}$$

where c is a scaling factor for the penalty. By penalizing unsafe actions, the reward replacement module influences the cumulative reward estimation, enabling the agent to learn policies that avoid unsafe actions and associated high ramp occupancy states.

4. Experiment

We conduct secondary development based on the microscope traffic simulation software Simulation of Urban Mobility (SUMO) [33]. Through the traffic control interface (Traci), we implement the control action provided by the RL agent on the traffic lights and obtain the detection data from various detectors. The collected data from each simulation step is integrated to form the state of the environment for the control step. The integrated state is used as the input of the neural network. Due to the large computational complexity of microscopic traffic simulation, the vectorized environment is employed to accelerate the collection speed of the trajectory data. A vectorized environment simultaneously opens multiple simulation environments to interact with one agent. These simulation environments are independent of each other, so we can parallelly process them. The interaction between the agent and the vectorized environment is shown in Figure 3.

In the previous section, we discussed how traffic flow at bottlenecks is highly susceptible to perturbations and may break down when approaching capacity. To enhance the realism of the experiments, we employ a two-step calibration based on the genetic algorithm [34]. The simulation road networks are divided into segments, each ranging from 300 to 500 m in length. The objective function of the genetic algorithm is defined as the mean absolute percentage error (MAPE) between the simulated segment flow and real flow measured over 5-minute intervals. The calibration process is carried out in two stages: First, we initially calibrate the simulation parameters one at a time to identify the set of parameters that have significant impacts on the objective function. Second, the identified parameters are then jointly optimized using the genetic algorithm to determine their optimal combination. Following this calibration process, the MAPE of the flow is 9.6%, indicating a high degree of alignment between the simulation and real-world observations. A summary of the calibration results is provided in Table 2.

4.1. Local Ramp Scenario. We choose the entrance ramp of Shiyang Road and its upstream and downstream sections on the Nanjing Ring Freeway as the local ramp simulation scenario. The total length of the simulated section is approximately 1.34 km, with an upstream section of 755 m, a merge area of 95 m, a downstream section of 486 m, and an on-ramp of 146 m. The on-ramp can accommodate a maximum of 42 vehicles. The mainline has 4 lanes, the on-ramp has 2 lanes, and the merge area has 5 lanes. The mainline has a lane drop bottleneck at the end of the merge area. The satellite map of the simulated section is shown in Figure 4. In order to eliminate the impact of various vehicle loading methods on the loop detectors, a virtual section has been added at the starting points of the mainline and the on-ramp, respectively.

The capacity of the mainline is 8280 veh/h, while the capacity of the ramp is 1930 veh/h. Based on the above capacity, we conducted experiments using the flow input shown in Figure 5. The modified cumulative arrival curve of the upstream and downstream of the bottleneck is shown in Figure 6; we rescaled it by subtracting the background flow, which is 8160 veh/h. To align the curves of the upstream and downstream, the two curves of the downstream are shifted to the left by the time it takes for vehicles to travel from the upstream to the downstream, which is 16 and 25 s, respectively. From 600 to 800 s, the outflow (slope of the curve) downstream shows a significant decrease compared to 500–550 s, resulting in the accumulation of vehicles between the detectors. Accompanied by the queue of vehicles, the significant divergence between the upstream and downstream curves can be observed in the 600 s, which indicates the existence of the capacity drop. Similarly, the divergence can also be found at 950 s, which means another capacity drop occurs during 950–1600 s.

In this scenario, we use PI-ALINEA as the baseline strategy as follows:

$$\begin{aligned} r(k) = r(k-1) + k_r \times \left(\widehat{O} - O_{\mathrm{out}}(k)\right) - k_p \\ \times \left[O_{\mathrm{out}}(k) - O_{\mathrm{out}}(k-1)\right], \end{aligned} \tag{16}$$

where k_r and k_p are two positive regulator parameters. So, we need to tune two parameters and calibrate the critical occupancy. We use grid search to find the optimal parameter combination, and the search scope of k_r is $[10, 100]$ and the search step is 10, while the search of k_p is $[0, 90]$ and the search step is also 10. The optimal parameter combination is $k_r = 90, k_p = 10$. To calibrate the critical occupancy, we plot the variation of the downstream occupancy within the range

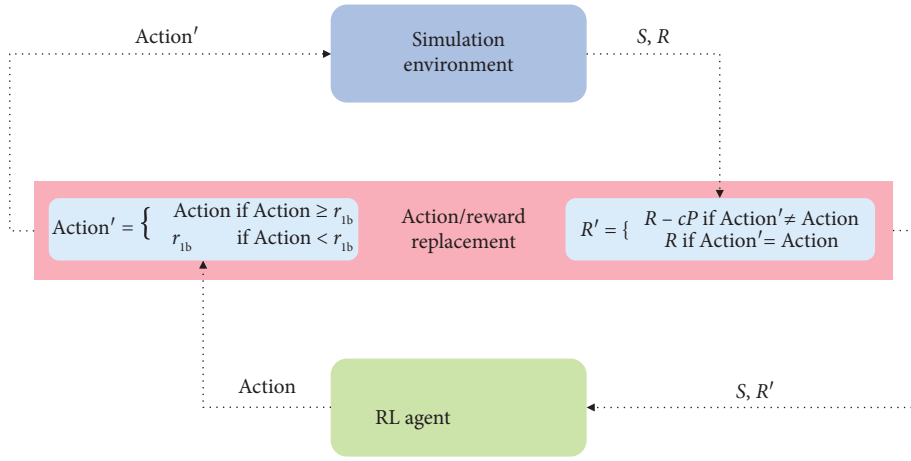

Figure 2: Learning process with lower bound constraint.

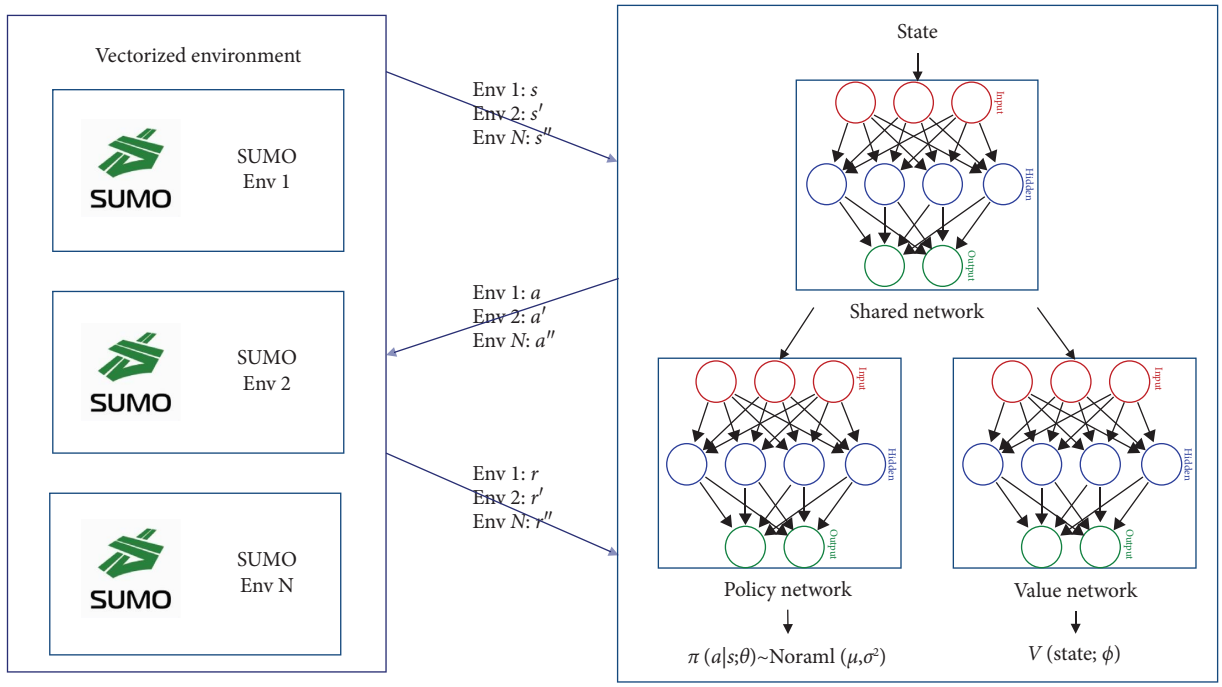

Figure 3: Interaction between agent and vectorized environment.

of 7400–9400 veh/h for the mainline and 600–1600 veh/h for the ramp. Figure 7 reveals that the downstream occupancy fluctuates in the range of 12%-13% when the mainline and ramp flow are relatively low (blue area in the bottom left corner). As the ramp input flow steadily increases, the downstream occupancy rises to around 14% (deep red area). However, as the mainline flow approaches its capacity, even a small ramp flow can significantly disrupt the mainline flow and lead to breakdown (light red/blue area in the middle) resulting in a decrease in downstream occupancy. To sum up, we choose 14% as the critical occupancy for the downstream bottleneck.

During the RL training process, the interaction between the agent and the environment is terminated if the queue length exceeds 90% of the ramp storage capacity. Every simulation step represents one second in the real world. Each control step is 15 s, that is 15 simulation steps, and the maximum number of interactions in a single episode simulation is 240 control steps. We select the TTS as the evaluation metric for control performance.

The TTS of the no-control strategy, ALINEA, and RL–based strategy are 595.77, 578.27, and 564.98 h, respectively. Compared to the no-control strategy, ALINEA reduces the travel time by 17.53 h, while the RL–based strategy reduces travel time by 30.79 h.

To verify the effect of the lower bound of the metering rate in preventing ramp spillback, we retrain the agent by introducing the action and reward substitution modules while maintaining the same neural network structure and other parameters. The comparison of episode interaction

Table 2: Simulation parameter values.

Parameter	Value
departLane	Best
departPos	Free
departSpeed	Random
speedFactor	1.0
speedDev	0.03
minGap	2.0
Accel	2.6
Decel	4.5
Sigma	0.3
tau	1.1
lcCooperative	1.0
lcSpeedGain	2.5
lcImpatience	1.0
lcOvertakeRight	0.3
lcLookaheadLeft	0.5
lcAssertive	3.0
lcStrategic	0.8

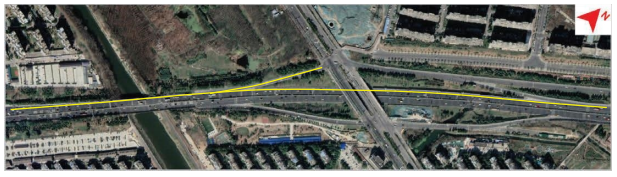

Figure 4: Satellite map of the local ramp scenario.

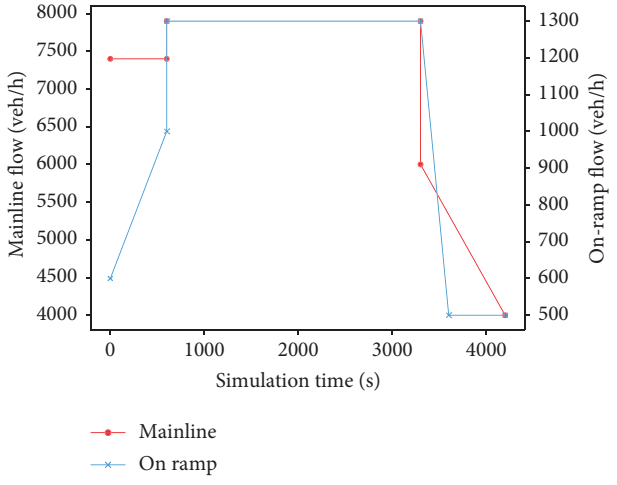

Figure 5: Local ramp input flow.

steps and the network TTS during the training process with/without the lower bound of metering rate constraint are shown in Figures 8 and 9. We can find that the neural network converges around 5×10^4 simulation steps when the lower bound constraint is not introduced. In this case, the agent can complete the entire 240 control steps interaction in most episodes, however, despite the convergence of the neural network, the spillback of the ramp still occurs (see the burrs of the orange curve). This phenomenon is due to the fact that the agent selects a series of ramp metering rates that violate the lower bound constraint,

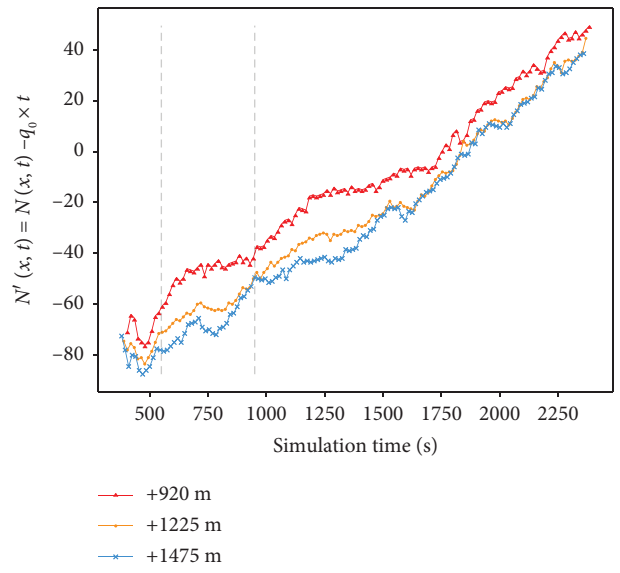

Figure 6: The cumulative curves at different locations.

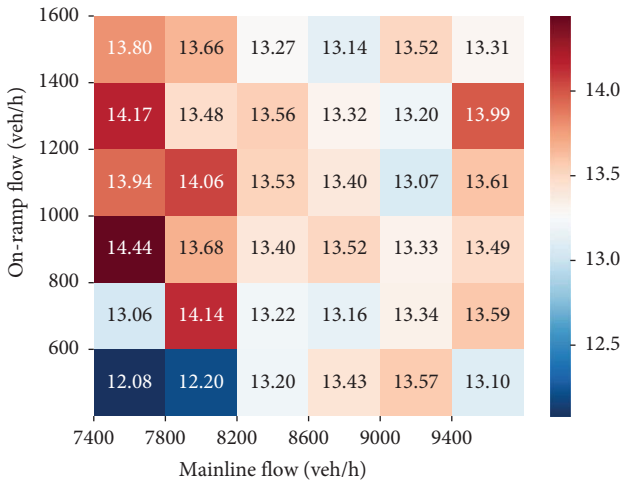

Figure 7: Variation of the downstream occupancy.

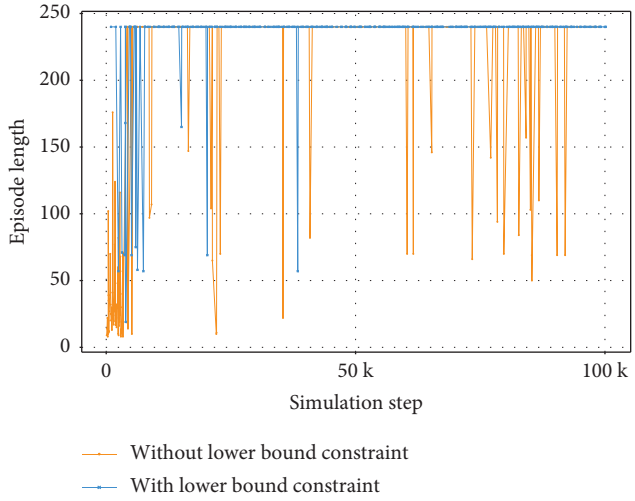

Figure 8: Episode interaction step comparison for local ramp scenario (first 100 k step).

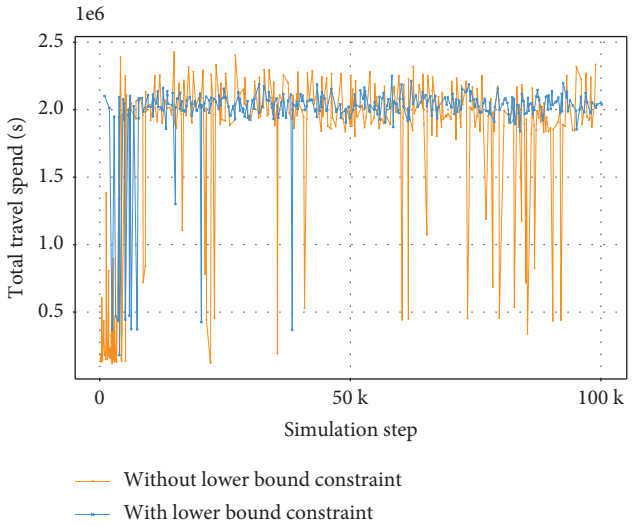

Figure 9: Network TTS comparison for local ramp scenario (first 100 k step).

resulting in high queue length on the ramp and triggering the premature termination of interaction. After introducing the lower bound constraint, we find that such a constraint does not harm the strategy performance, and the convergence speed does not have significant changes. However, the training process becomes more stable, there seldom occurs the spillback (the blue curve has fewer burrs). From Figure 10, we can find that the percentage of action replacements decreases during the training progresses, which indicates that introducing lower bound constraints during training can effectively reduce the number of unsafe actions. In addition, the reward substitution module can guide the agent to proactively avoid the ramp spillback state.

4.2. Multiramp Scenario. We select the section from Maqun South Road on-ramp to the Inner Ring South Line on-ramp of Nanjing Ring Freeway as the multiramp simulation scenario. The total length of the simulated section is approximately 7.64 km, including four on-ramps: Maqun South Road, Shuangqi Road, Shiyang Road, and Inner Ring South Line as well as four corresponding off-ramps. The simulated segment has multiple bottlenecks, the downstream of Maqun South Road and Inner Ring South Line has lane drop bottlenecks, and Shuangqi Road and Shiyang Road on-ramp are located close to their downstream off-ramps, with distances of 517 and 397 m, respectively, forming two weaving area. The satellite map is shown in Figure 11.

The capacity of the mainline is 8100 veh/h, and the capacity of the on-ramps are 2540, 2650, 1930, and 2300 veh/h. Peak hour downstream traffic flow for each on-ramp is shown in Figure 12.

The modified cumulative arrival curves at the upstream and downstream of each ramp are shown in Figure 13. Similar to the local ramp scenario, varying degrees of capacity drop can be observed at each downstream bottleneck of the on-ramp.

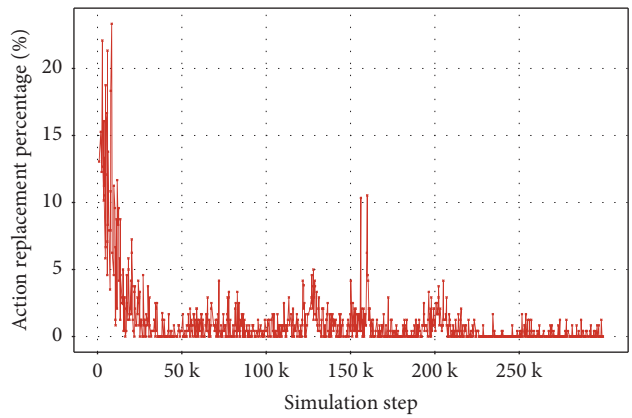

Figure 10: Variation of action replacement percentage for local ramp scenario.

Figure 11: Satellite map of the multiramp scenario.

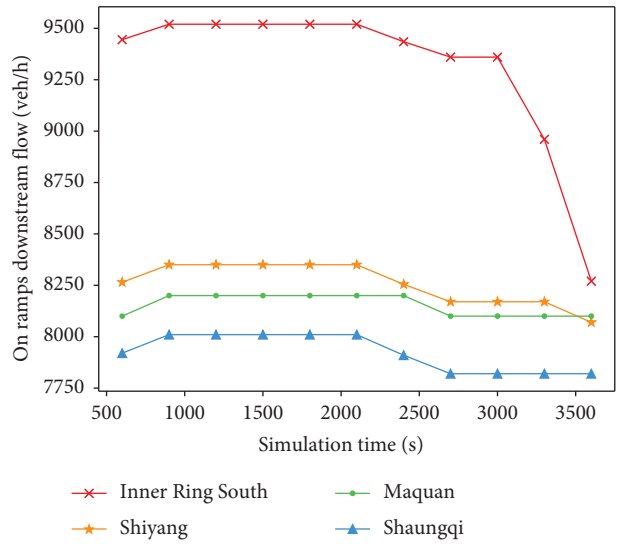

Figure 12: Downstream traffic flow for each on-ramp.

In this set of experiments, the maximum interaction control step is 240 per episode, and we use the grid search method similar to the local ramp scenario to find the optimal parameter combinations for PI-ALINEA and HERO. The optimal (k_r, k_p) for each ramp are $(110, 20)$, $(100, 0)$, $(100, 0)$, and $(100, 20)$, respectively, and the critical occupancy for each ramp is 12.3%, 12.5%, 13.0%, and 13.5%, respectively. Considering the complexity of the road structure in the multiramp scenario, a decentralized structure would require a large number of shared states among agents. In addition, the proximity of the Shuangqi Road ramp and the Shiyang Road ramp can lead to mutual interference and pose challenges for training. In this scenario, the dimension of the environment state is relatively small, and the

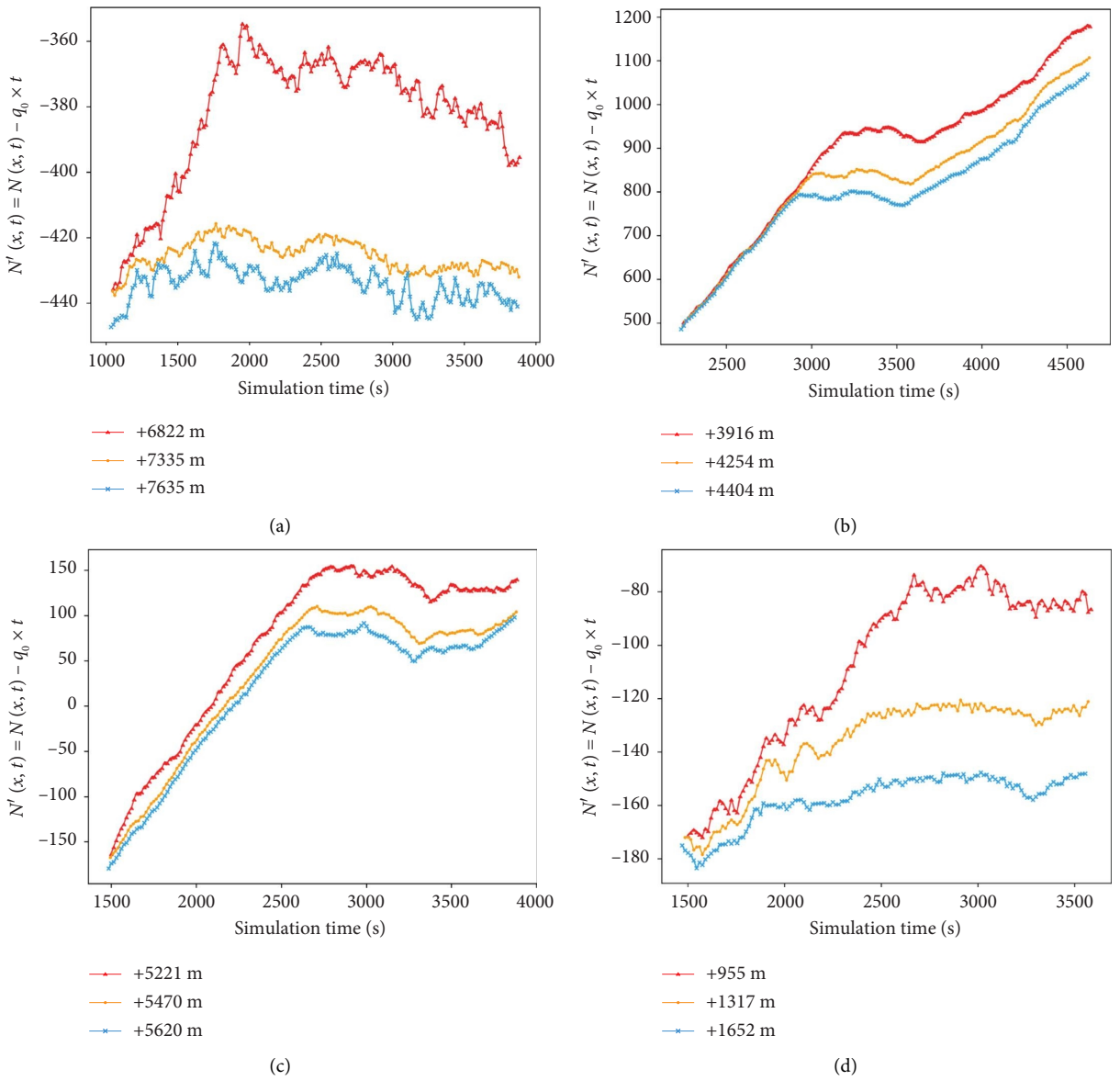

Figure 13: Modified cumulative arrival curves of multiramps: (a) Maqun South Road, (b) Shuangqi road, (c) Shiyang road, and (d) Inner Ring South Line.

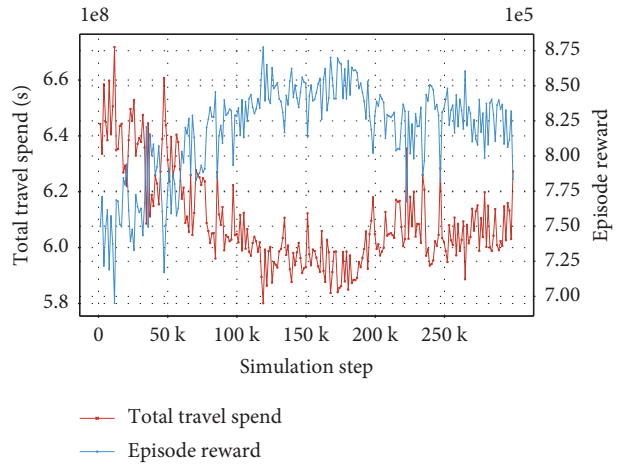

Figure 14: The variations of TTS and episode reward for experiment 1, multiramp scenario.

Table 3: TTS of all the tested metering strategies.

Strategies	TTS (h)	Average speed (m/s)	Delay on the adjacent network (h)
No-control	1901.43	15.67	20.35
ALINEA	1838.08	15.64	283.56
HERO	1756.70	16.10	447.69
RL–based	1630.45	16.45	354.52

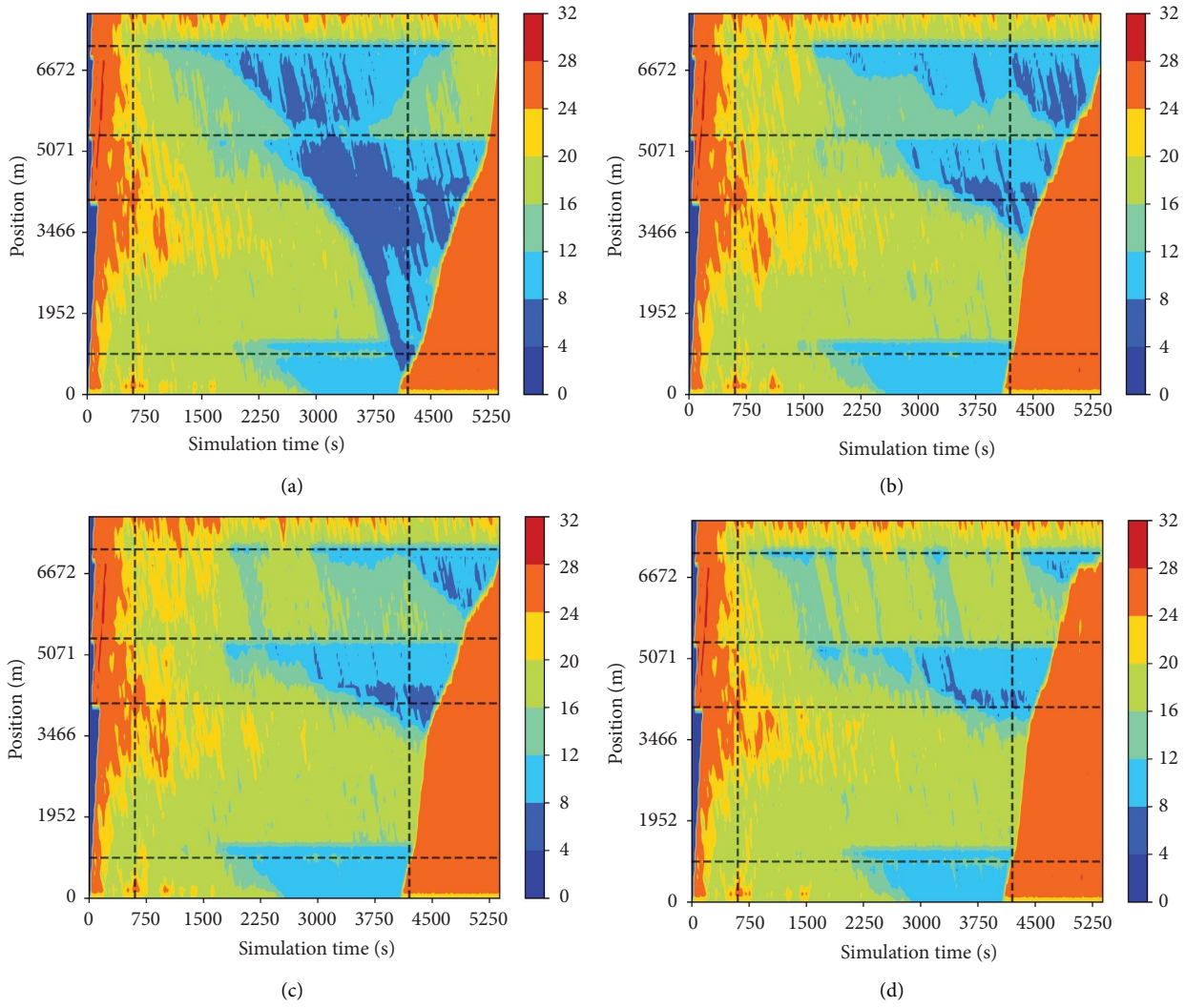

Figure 15: Average speed (m/s) under (a) no-control, (b) ALINEA, (c) HERO, and (d) RL–based policy.

scalability factors are not a concern; to simplify the implementation, we choose the centralized structure in the following experiments which means that a single agent will control all four on-ramp traffic lights.

In this scenario, two sets of experiments were conducted to evaluate the effectiveness of the proposed method. In the first experiment, no constraints are placed on the maximum queue length during the control process. The performance of four strategies, no-control strategy, ALINEA, HERO, and RL–based strategy, incorporating a queue penalty term on reward function is compared. This experiment is designed to demonstrate that RL–based ramp metering strategies, despite their advantages, are insufficient to prevent spillback under near-capacity demand conditions. In the second experiment, the maximum queue length for each on-ramp is strictly limited during interactions. If the spillback occurs, the interaction between the agent and environment immediately terminates, allowing the number of spillback events to be indirectly observed from the episode lengths during the training process. This setup is aimed at comparing the proposed method against the conventional method in terms of its ability to avoid spillback.

4.2.1. Experiment 1: Allow Spillback. The variation of episode TTS and cumulative rewards during the training process is shown in Figure 14. The TTS and other metrics of all the tested metering strategies are summarized in Table 3.

From the Table 3, we can observe that all three control strategies can effectively reduce the TTS, with the RL–based strategy performing the best, followed by HERO. To provide a detailed comparison of the differences between these strategies, Figure 15 shows the variation of average speed over time among all cross-sections where loop detectors are installed under various control strategies. From top to bottom, left to right, the subfigures are no-control strategy, ALINEA, HERO, and RL–based strategy, and the horizontal dashed line indicates the location of the on-ramps, while the vertical dashed line represents the peak input flow duration.

The no-control strategy subfigure shows that the congestion starts to form downstream around 1500 s, and the congestion gradually spreads from the downstream of the Inner Ring South Line to the Maqun South Road from 2500 s, forming large-scale congestion in the weaving area between Shuangqi Road and Shiyang Road. The ALINEA strategy effectively reduces the queue length located upstream of each ramp. The HERO strategy delays the start time of the downstream queue by 1500 s and further reduces the queue length compared to ALINEA. The RL–based strategy completely eliminates the congestion occurring upstream of the Inner Ring South Line and avoids the queue of Shiyang Road spreading to Shuangqi Road.

Although the RL–based method achieves over a 10% reduction in total travel time, it simultaneously causes significant delays, amounting to hundreds of hours on adjacent road networks. This outcome highlights that simply modifying the reward function is inadequate to prevent ramp queue spillback effectively. To further evaluate the proposed method's ability to address this issue, the next experiment compares its training process with that of conventional RL–based ramp metering.

4.2.2. Experiment 2: Forbid Spillback. The comparison of the episode TTS and interaction steps during the training process with/without the metering rate lower bound constraint is shown in Figures 16 and 17.

When ramp queue spillback is forbidden, the RL ultimately converges to a different policy compared to Experiment 1. Such divergence can be attributed to the relatively high peak input flow, which is approximately 13.25% higher than the mainline capacity, exceeding the maximum adjustment range under the limited queue length of the ramps. For comparison, all control strategies in Experiment 1 cause spillback. Therefore, it is reasonable to yield different results for this experiment. From Figures 16 and 17, we find that the lower bound constraint of the ramp metering rate does not have a negative impact on the performance and convergence speed, which is similar to Experiment 1. The action substitution module can significantly reduce the number of spillback events during the training process. The proportion of replaced action decreased from the initial 22.86% to less than 1% at the end of training as shown in Figure 18,

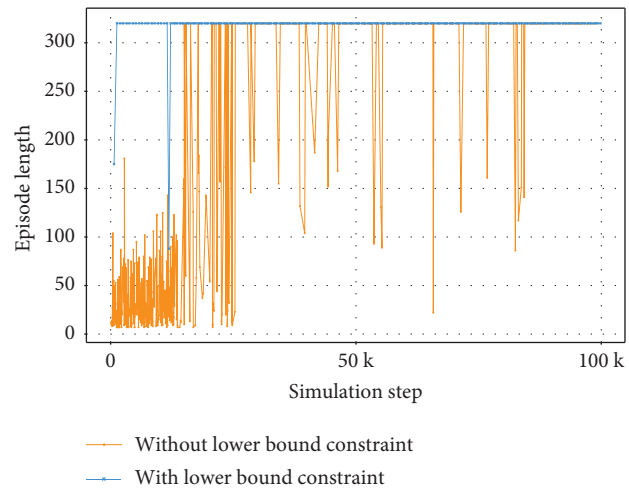

Figure 16: Episode interaction step comparison for experiment 2, multiramp scenario (first 100 k step).

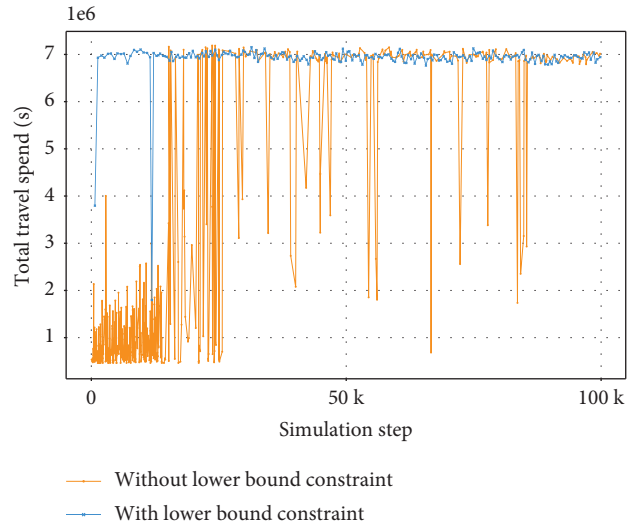

Figure 17: Network TTS comparison for experiment 2, multiramp scenario (first 100 k step).

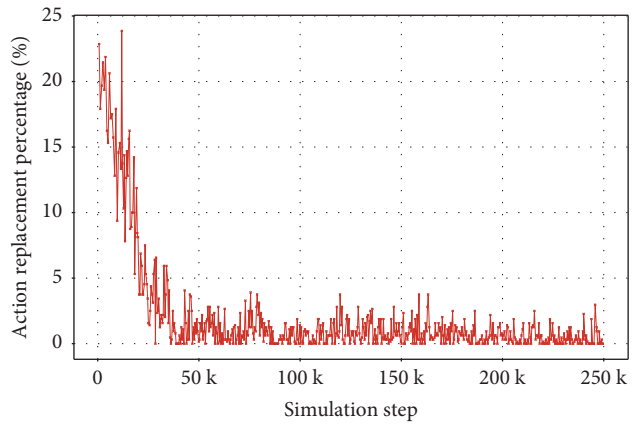

Figure 18: Variation of action replacement percentage for experiment 2, multiramp scenario (first 100 k step).

indicating that the reward substitution module guides the agent to learn the on-ramp storage capacity limitation and consequently, avoid the risk of spillback.

5. Conclusions

This paper introduces the action replacement module to the training process of the RL–based ramp metering strategy. We demonstrate the effectiveness of this method through a series of comparative experiments. The results show that with the introduction of the lower bound constraint, the action replacement module significantly reduces the number of spillback events, thereby improving the influence of RL–based ramp metering algorithms on adjacent road networks during online training. This enhancement suggests that our method is promising for facilitating the training and deployment of corresponding control strategies in real-world environments. Moreover, given that our approach is not dependent on any specific RL algorithm, it can be paired with various suitable RL techniques in different scenarios, such as variable speed limit control. By integrating our approach, these strategies are expected to reduce the negative impact of policy exploration on traffic efficiency and safety, thereby enhancing their practicality in real-world applications.

For future work, we will consider the multiagent RL–based method to solve the scalability issues of the current method and extend the application of the action replacement module to other advanced RL paradigms such as offline RL and meta-learning methods. These advancements could further enhance the robustness and applicability of the proposed approach in diverse traffic scenarios. Furthermore, integrating additional traffic control measures, such as variable speed limits may also be a promising research direction.

Data Availability Statement

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This research was supported by the National Natural Science Foundation of China (Grant no. 52131203) and the Fundamental Research Funds for the Central Universities.

References

[1] S. Trubia, S. Curto, S. Barberi, A. Severino, F. Arena, and G. Pau, "Analysis and Evaluation of Ramp Metering: From Historical Evolution to the Application of New Algorithms and Engineering Principles," *Sustainability* 13, no. 2 (2021): 850, https://doi.org/10.3390/su13020850.

[2] J. C. Aydos and A. O. Brien, "SCATS Ramp Metering: Strategies, Arterial Integration and Results," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (2014), 2194–2201, https://doi.org/10.1109/ITSC.2014.6958028.

[3] M. Papageorgiou, H. Hadj-Salem, and J.-M. Blosseville, "ALINEA: A Local Feedback Control Law for On-Ramp Metering," *Transportation Research Record* no. 1320 (1991): https://trid.trb.org/View/365587.

[4] Y. Wang, E. B. Kosmatopoulos, M. Papageorgiou, and I. Papamichail, "Local Ramp Metering in the Presence of a Distant Downstream Bottleneck: Theoretical Analysis and Simulation Study," *IEEE Transactions on Intelligent Transportation Systems* 15, no. 5 (2014): 2024–2039, https://doi.org/10.1109/TITS.2014.2307884.

[5] E. Smaragdis and M. Papageorgiou, "Series of New Local Ramp Metering Strategies: Emmanouil Smaragdis and Markos Papageorgiou," *Transportation Research Record: Journal of the Transportation Research Board* 1856, no. 1 (2003): 74–86, https://doi.org/10.3141/1856-08.

[6] J. R. D. Frejo and B. De Schutter, "Feed-Forward ALINEA: A Ramp Metering Control Algorithm for Nearby and Distant Bottlenecks," *IEEE Transactions on Intelligent Transportation Systems* 20, no. 7 (2019): 2448–2458, https://doi.org/10.1109/TITS.2018.2866121.

[7] I. Papamichail and M. Papageorgiou, "Traffic-Responsive Linked Ramp-Metering Control," *IEEE Transactions on Intelligent Transportation Systems* 9, no. 1 (2008): 111–121, https://doi.org/10.1109/TITS.2007.908724.

[8] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model Predictive Control for Optimal Coordination of Ramp Metering and Variable Speed Limits," *Transportation Research Part C: Emerging Technologies* 13, no. 3 (2005): 185–209, https://doi.org/10.1016/j.trc.2004.08.001.

[9] Y. Han, M. Ramezani, A. Hegyi, Y. Yuan, and S. Hoogendoorn, "Hierarchical Ramp Metering in Freeways: An Aggregated Modeling and Control Approach," *Transportation Research Part C: Emerging Technologies* 110 (2020): 1–19, https://doi.org/10.1016/j.trc.2019.09.023.

[10] S. Heshami and L. Kattan, "Ramp Metering Control under Stochastic Capacity in a Connected Environment: A Dynamic Bargaining Game Theory Approach," *Transportation Research Part C: Emerging Technologies* 130 (2021): 103282, https://doi.org/10.1016/j.trc.2021.103282.

[11] Y. Han, M. Wang, and L. Leclercq, "Leveraging Reinforcement Learning for Dynamic Traffic Control: A Survey and Challenges for Field Implementation," *Communications in Transportation Research* 3 (2023): 100104, https://doi.org/10.1016/j.commtr.2023.100104.

[12] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert Level Control of Ramp Metering Based on Multi-Task Deep Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems* 19, no. 4 (2018): 1198–1207, https://doi.org/10.1109/TITS.2017.2725912.

[13] B. Liu, Y. Tang, Y. Ji, Y. Shen, and Y. Du, "A Deep Reinforcement Learning Approach for Ramp Metering Based on Traffic Video Data," *Journal of Advanced Transportation* 2021, no. 1 (2021): 1–13, https://doi.org/10.1155/2021/6669028.

[14] Y. Han, M. Wang, L. Li, C. Roncoli, J. Gao, and P. Liu, "A Physics-Informed Reinforcement Learning-Based Strategy for Local and Coordinated Ramp Metering," *Transportation Research Part C: Emerging Technologies* 137 (2022): 103584, https://doi.org/10.1016/j.trc.2022.103584.

[15] C. Wang, Y. Xu, J. Zhang, and B. Ran, "Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning," *IEEE Transactions on Intelligent*

Transportation Systems 23, no. 9 (2022): 15522–15535, https://doi.org/10.1109/TITS.2022.3141730.

[16] M. Davarynejad, A. Hegyi, J. Vrancken, and J. van den Berg, "Motorway Ramp-Metering Control With Queuing Consideration Using Q-Learning," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (2011), 1652–1658, https://doi.org/10.1109/ITSC.2011.6082976.

[17] C. Lu and J. Huang, "A Self-Learning System for Local Ramp Metering With Queue Management," *Transportation Planning and Technology* 40, no. 2 (2017): 182–198, https://www.tandfonline.com/doi/abs/10.1080/03081060.2016.1266166.

[18] F. Deng, J. Jin, Y. Shen, and Y. Du, "A Dynamic Self-Improving Ramp Metering Algorithm Based on Multi-Agent Deep Reinforcement Learning," *Transportation Letters* 16, no. 7 (2024): 649–657, https://doi.org/10.1080/19427867.2023.2231638.

[19] J. Cheng, C. Ye, N. Wang, et al., "A Deep Reinforcement Learning Based Ramp Metering Control Method Considering Ramp Outflow∗," *IFAC-PapersOnLine* 58, no. 10 (2024): 200–205, https://doi.org/10.1016/j.ifacol.2024.07.340.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained Policy Optimization," in *Proceedings of the 34th International Conference on Machine Learning* (PMLR, 2017), 22–31, https://proceedings.mlr.press/v70/achiam17a.html.

[21] G. Dalal, K. Dvijotham, M. Vecerik, H. Todd, C. Paduraru, and Y. Tassa, "Safe Exploration in Continuous Action Spaces," *arXiv* (2018): https://doi.org/10.48550/arXiv.1801.08757.

[22] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without Error: Towards Safe Reinforcement Learning via Human Intervention," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018), 2067–2069.

[23] Y. Xu, Z. Liu, G. Duan, J. Zhu, X. Bai, and J. Tan, "Look Before You Leap: Safe Model-Based Reinforcement Learning With Human Intervention," in *Proceedings of the 5th Conference on Robot Learning* (PMLR, 2022), 332–341, https://proceedings.mlr.press/v164/xu22a.html.

[24] S. Oh and H. Yeo, "Impact of Stop-and-Go Waves and Lane Changes on Discharge Rate in Recovery Flow," *Transportation Research Part B: Methodological* 77, no. July (2015): 88–102, https://doi.org/10.1016/j.trb.2015.03.017.

[25] A. Srivastava and N. Geroliminis, "Empirical Observations of Capacity Drop in Freeway Merges With Ramp Control and Integration in a First-Order Model," *Transportation Research Part C: Emerging Technologies* 30 (2013): 161–177, https://doi.org/10.1016/j.trc.2013.02.006.

[26] K. Chung, J. Rudjanakanoknad, and M. J. Cassidy, "Relation Between Traffic Density and Capacity Drop at Three Freeway Bottlenecks," *Transportation Research Part B: Methodological* 41, no. 1 (2007): 82–95, https://doi.org/10.1016/j.trb.2006.02.011.

[27] M. J. Cassidy and R. L. Bertini, "Some Traffic Features at Freeway Bottlenecks," *Transportation Research Part B: Methodological* 33, no. 1 (1999): 25–42, https://doi.org/10.1016/S0191-2615(98)00023-X.

[28] M. J. Cassidy and J. Rudjanakanoknad, "Increasing the Capacity of an Isolated Merge by Metering Its On-Ramp," *Transportation Research Part B: Methodological* 39, no. 10 (2005): 896–913, https://doi.org/10.1016/j.trb.2004.12.001.

[29] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in Neural Information Processing Systems*, 12 (MIT Press, 1999), https://proceedings.neurips.cc/paper_files/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html.

[30] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust Region Policy Optimization," *arXiv* (2017): https://doi.org/10.48550/arXiv.1502.05477.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv* (2017): https://doi.org/10.48550/arXiv.1707.06347.

[32] Y. Li, X. Li, J. Qiao, and C. Zhang, "Optimal Design of Bimodal Hierarchical Transit Systems: Tradeoffs Between Costs and CO2 Emissions," *Research in Transportation Economics* 109 (2025): 101496, https://doi.org/10.1016/j.retrec.2024.101496.

[33] P. A. Lopez, E. Wiessner, M. Behrisch, et al., "Microscopic Traffic Simulation Using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018), 2575–2582, https://doi.org/10.1109/ITSC.2018.8569938.

[34] S. Katoch, S. S. Chauhan, and V. Kumar, "A Review on Genetic Algorithm: Past, Present, and Future," *Multimedia Tools and Applications* 80, no. 5 (2021): 8091–8126, https://doi.org/10.1007/s11042-020-10139-6.