Check for updates

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–12
© National Academy of Sciences:
Transportation Research Board 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0361198120935873
journals.sagepub.com/home/trr
SAGE

Cooperative Highway Work Zone Merge Control Based on Reinforcement Learning in a Connected and Automated Environment

Tianzhu Ren[1], Yuanchang Xie[2], and Liming Jiang[2]

Abstract

Given the aging infrastructure and the anticipated growing number of highway work zones in the U.S.A., it is important to investigate work zone merge control, which is critical for improving work zone safety and capacity. This paper proposes and evaluates a novel highway work zone merge control strategy based on cooperative driving behavior enabled by artificial intelligence. The proposed method assumes that all vehicles are fully automated, connected, and cooperative. It inserts two metering zones in the open lane to make space for merging vehicles in the closed lane. In addition, each vehicle in the closed lane learns how to adjust its longitudinal position optimally to find a safe gap in the open lane using an off-policy soft actor critic reinforcement learning (RL) algorithm, considering its surrounding traffic conditions. The learning results are captured in convolutional neural networks and used to control individual vehicles in the testing phase. By adding the metering zones and taking the locations, speeds, and accelerations of surrounding vehicles into account, cooperation among vehicles is implicitly considered. This RL-based model is trained and evaluated using a microscopic traffic simulator. The results show that this cooperative RL-based merge control significantly outperforms popular strategies such as late merge and early merge in terms of both mobility and safety measures. It also performs better than a strategy assuming all vehicles are equipped with cooperative adaptive cruise control.

Bottlenecks generated by work zones as well as traffic incidents are among the most important contributors to non-recurring congestion and secondary crashes. Many previous work zone studies focused on merge control and proposed a variety of strategies such as early merge (EM) (*1*) and late merge (LM) (*2*) to improve work zone throughput. EM typically uses a sequence of "Do Not Pass" signs that can be activated/deactivated depending on traffic to create a no-passing zone of varying length. A traffic sensor is mounted on each sign to monitor traffic in the open lane. The purpose of the no-passing zone is to encourage drivers in the closed lane to switch to the open lane before reaching the end of the dynamically changing queue (or slow-moving traffic) to improve safety and efficiency. EM often creates high-speed but low-density flow at the merging point. While for LM, drivers in both open and closed lanes are urged to stay in their respective lanes until the merging point, where they take turns to merge. Compared with EM, LM can effectively reduce the overall queue length, since both lanes are used for queue storage. However, LM often generates low-speed but high-density flow at the merging point. Ideally, the best merge control should result in high-speed and high-density flow.

Some advanced driving assistant systems, such as adaptive cruise control (ACC) (*3*), enable vehicles to drive at a high speed while maintaining a small gap (i.e., high density). Such a feature is only intended for improving vehicle longitudinal control and cannot address the challenging work zone merge problem. Also, an ACC-equipped vehicle only considers its interactions with the vehicle immediately in front of it and in the same lane (including vehicles attempting to merge into its lane), trying to make optimal decisions locally. To improve work zone traffic operations, it is important for individual vehicles to take global traffic conditions into

[1]Amazon, Seattle, WA
[2]Department of Civil and Environmental Engineering, University of Massachusetts Lowell, Lowell, MA

Corresponding Author:
Yuanchang Xie, Yuanchang_Xie@uml.edu

consideration and cooperate with other vehicles in both the open and closed lanes.

To enable collaborative driving behavior among connected and automated vehicles (CAV), there are three main challenges: (a) how to take the vast amount of unstructured traffic information into consideration effectively; (b) how to choose an optimal control policy based on the dynamically changing surrounding traffic that maximizes the benefits of the subject vehicle in the long run instead of just the next few time steps; and (c) how to make the best decisions based not only on the subject merging vehicle's state, but also on its surrounding vehicles' current states and possible moves in the future. Some previous studies have attempted to address the collaborative merge problem. Chen et al. (*4*) applied a gap acceptance algorithm and proposed several rules to decide a vehicle's actions before merging into the target lane. Urmson et al. (*5*) used a slot-based approach for cooperative merging control. Van et al. (*6*) proposed a cooperative adaptive cruise control (CACC) to allow CAVs to drive in a cooperative manner. However, these rule-based methods depend heavily on specific situations which are pragmatically vulnerable due to their inability to adapt to unforeseen environments.

Reinforcement learning (RL) has been successfully applied to a variety of fields with the growing availability of cost-effective high-performance computing hardware. RL together with deep neural networks can take large dimensions of state space into consideration, making it very appealing for work zone control. Using RL, analysts do not need to specify explicitly how a work zone changes from one state into another (i.e., state transition probability matrix), which dramatically reduces the modeling effort needed, particularly the trouble associated with specifying the uncertain state transition probability matrix. Vehicle agents can learn from a huge number of simulated scenarios about the complex nonlinear relationship between their next moves and work zone traffic operations, and find actions with the maximum long-term reward.

Due to these desirable features, RL has been applied in self-driving vehicles such as NVIDIA (*7*), Tesla Autopilot (*8*), and Google Waymo. RL has also been adopted for CACC (*9, 10*), and it has been demonstrated that RL can lead to safe and efficient longitudinal control in a connected vehicles environment. Additionally, RL-based vehicle controllers have been tested and validated in both simulated environments and real-world experiments (*11*). However, no merging maneuvers were considered in these studies. Several researchers have proposed RL approaches to tackle the problem of on-ramp merge control (*12, 13*). Although similar, on-ramp and work zone controls are fundamentally different. For example, for work zone control it is ideal for vehicles in both the open and closed lanes to travel at approximately the same speed and flow rate, while for on-ramp control ramp vehicles usually need to yield to highway mainline traffic. Some researchers have also applied RL in ramp metering (*14, 15*). Specifically, Fares et al. (*14*) developed a RL model to optimally control the density of freeway mainstream for maximizing traffic throughput and minimizing travel time. Their model was formulated as a Markov decision process (*16*) and solved by Q-learning (*17*). Yang et al. (*15*) proposed a deep Q-network (DQN) (*18*) control strategy to identify the optimal ramp metering rate. The DQN considered upstream and downstream traffic volumes as the input state and chose either green or red for the ramp meter traffic light as the action at each decision interval. Yu et al. (*19*) applied deep Q-learning to control a simulated car for turning and obstacle avoidance maneuvers. These studies all considered a discrete action space due to its simplicity and fast convergence, although many vehicle control problems [e.g., (*19*)] very likely may benefit more from using a continuous action space. Sallab et al. (*20*) compared a discrete action-space DQN with a continuous action-space deep deterministic actor critic (DDAC) (*21*) for lane-keeping assistance based on an open source car simulator, and the results showed that the discrete DQN method led to abrupt steering maneuvers while the continuous DDAC method generated better performance and smoother control.

This research proposes a deep neural network-based RL control approach that guides automated vehicles (AVs) through work zones. Specifically, a work zone is divided into two metering zones and a merging zone (see Figure 1). In the metering zones, AVs are not allowed to change lanes and they focus on adjusting longitudinal positions using the proposed RL method. By the time AVs reach the merging/lane reduction point, they will be able to maintain a sufficient front gap if all vehicles were projected onto a single lane. In this way, they can merge safely and form a high-speed and high-density vehicle platoon. The key to this proposed approach is how to adjust AVs' longitudinal positions properly in the metering zones. In this research, each AV in the closed lane is considered as a RL agent. It learns the best control strategy through its interactions with the simulated traffic environment using VISSIM. At each time step, this agent takes an action (i.e., acceleration, deceleration). At the next time step, the value for its previous action is updated based on a set of reward functions and the interactions between the agent and the environment. To improve the control model's generalization ability, a deep neural network is used to store the learning results. The proposed RL approach is detailed in the next section.

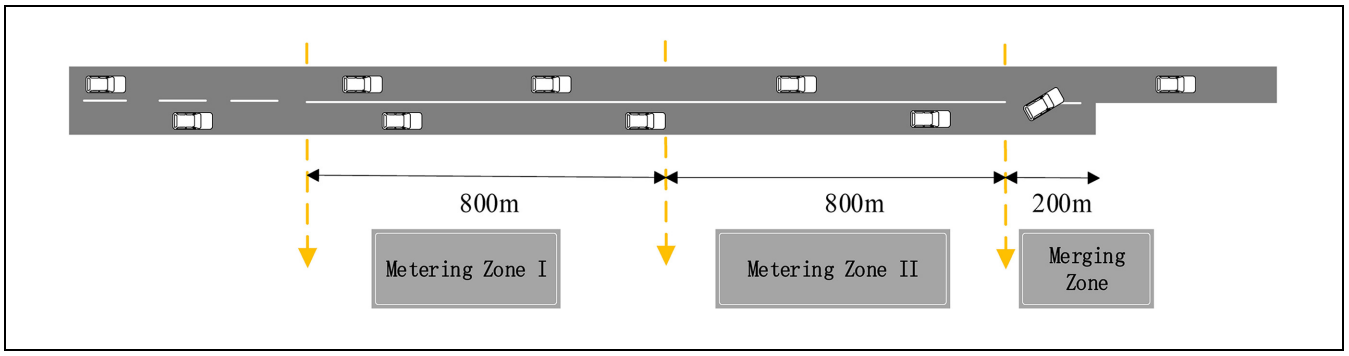

Figure 1. Overview of work zone model to evaluate reinforcement learning (RL)-based control.

Methodology

Overview

As shown in Figure 1, a work zone is divided into two metering zones followed by a merging zone. All vehicles approaching the work zone are instructed to increase their distance headways upon entering Metering Zone I. Specifically, each vehicle needs to increase its front distance headway to twice the safe distance needed for the corresponding speed (assuming 70 km/h). Metering Zone I is used to provide sufficient distance (i.e., reaction time) for vehicles to double their front gaps, and lane changing is prohibited in this zone. In Metering Zone II, vehicles in the open lane (left lane in Figure 1) will adopt the same car-following behavior as in Metering Zone I, while vehicles in the closed lane (right lane in Figure 1) are required to adjust their longitudinal positions. By the time vehicles reach the merging/lane reduction point, they will be able to maintain a sufficient front gap if all vehicles were projected onto a single lane. Following this longitudinal control strategy, toward the end of Metering Zone II, if vehicles in both lanes are projected onto a single virtual lane, all the distance headways are expected to be close to but greater than the minimum safe distance gap. In the Merging Zone, lane changes are allowed and vehicles in the two lanes take turns to merge. In summary, the core of the RL-based method is the longitudinal control in the two metering zones, where lane changes are prohibited. Before Metering Zone I, vehicles follow normal driving behavior. After Metering Zone II, vehicles also follow normal driving behavior other than being instructed to merge in the merging zone.

In this study, the deep neural network-based RL strategy and other benchmark strategies are evaluated using VISSIM microscopic traffic simulation. VISSIM provides a DriverModel_DLL (DLL stands for dynamic link library) interface that allows users to replace the default driving behavior models with custom-developed models. In Metering Zone II, vehicles in the right lane are controlled by a convolutional neural network trained by RL, and left-lane vehicles are controlled by a modified VISSIM default driving behavior model. The modification simply doubles the default time headway to create sufficient gaps for right-lane vehicles to merge in the Merging Zone.

Our initial goals included determining the optimal lengths (either dynamic or static values) for the two metering zones using RL, or letting individual vehicles decide where to merge depending on real-time traffic. However, this subtask alone turned out to be very challenging. In the current study, the lengths of the two zones are determined empirically based on VISSIM simulation to be 800 m. Their optimal values are not necessarily the same as in this study. It is anticipated that optimal lengths will further improve the performance of the proposed RL control. As a new merge control strategy focusing on vehicle cooperation, the proposed method can be expanded or improved in many directions. Therefore, the results reported in this study only set the floor for the performance of the proposed RL merge control.

Deep RL

In this section, a detailed description of the RL approach is provided, including the basics of RL, state representation, neural network architecture, and soft actor critic (*22*) RL, and reward shaping.

In this research, the control of right-lane vehicles (see Figure 1) is formulated as a Markov decision process consisting of numerous states s primarily defined by the surrounding traffic. Based on the learned policy π, an action a is selected at each state and executed. After the execution, the system (i.e., work zone traffic operations) will react to the action, from which a reward r can be observed, and transit to a new state s. The reward and the current and new states are then used to update the policy. To take each action's long-term reward into consideration, the expected discounted cumulative reward $\sum R$ is calculated along with the policy from the initial

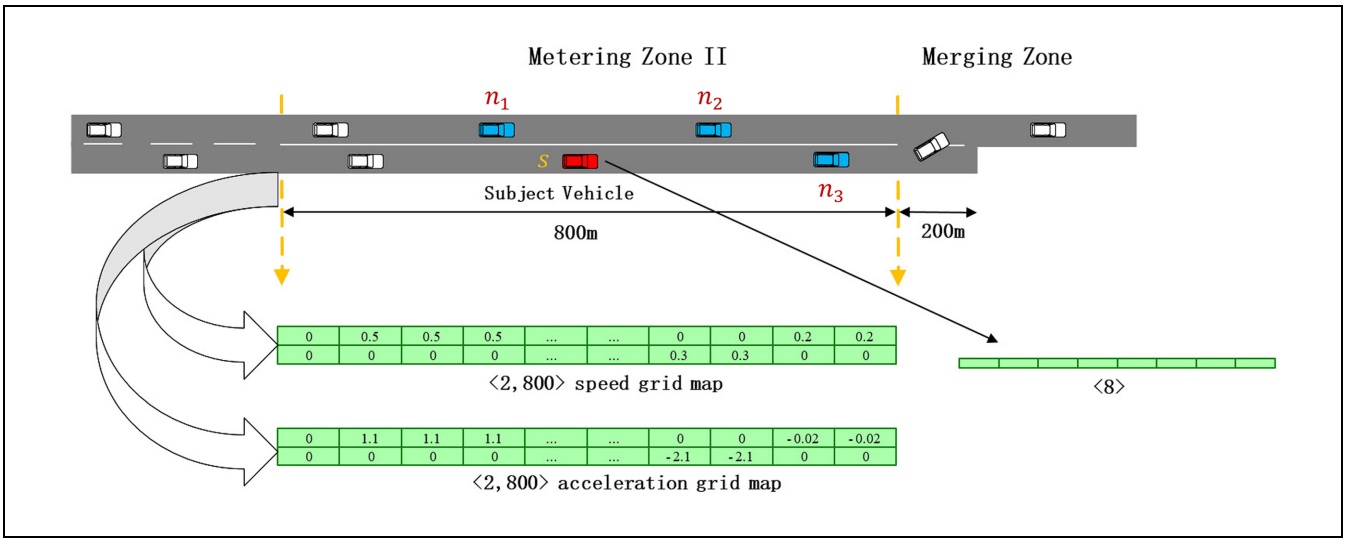

Figure 2. State representation for the proposed deep reinforcement learning algorithm.

state (a vehicle enters the work zone) to the terminal state (a vehicle merges into the open lane in the merging zone).

The RL results can be stored either by function approximation or in a table. The function approximation approach, such as deep deterministic policy gradient (DDPG) (*21*) and the proposed soft actor critic, employs mathematical models (e.g., deep neural networks) to approximate the value function and/or policy function, while the table approach, such as Q-learning, uses a table to store value or policy function values. Table based RL methods cannot handle the estimation of large Q tables (e.g., when the action space is continuous). Therefore, the function approximation approach is adopted in this study, which can handle well the curse of the dimensionality problem due to continuous state and action spaces and result in stable model performance in a highly complex environment (e.g., highway work zone traffic).

State Representation. In this study, the system state is defined by three components: network speed grid map, network acceleration grid map, and an eight-element vector (see explanation later in this section) representing the traffic surrounding the subject vehicle (i.e., the red vehicle in Figure 2) being controlled by RL.

As in Figure 2, the 800 m Metering Zone II is divided into 2×800 cells for the network speed grid map and network acceleration grid map. Each row is for a lane and each cell is for a 1 m segment. The numbers in each cell represent either the speed or the acceleration of the vehicle occupying that cell. If a vehicle occupies multiple cells, then the speed/acceleration values in the corresponding cells will be equal.

The speed values illustrated in Figure 2 are normalized based on the actual vehicle speeds and are bounded by 0 and 1. The normalization is done via dividing the original speed values by the maximum speed in the training and testing processes. Similarly, the acceleration values in Figure 2 are normalized using the maximum absolute value and are bounded by –1 and 1.

In addition to the two grid maps, an eight-element vector is included to describe the relationship between the subject vehicle s (the red vehicle in Figure 2) and three surrounding vehicles (n_1, n_2, and n_3 in Figure 2). Among the eight elements, three are for the longitudinal distance gaps between vehicles (s, n_1), (s, n_2), and (s, n_3); another three are for the longitudinal speed differences (relative speeds) between vehicles (s, n_1), (s, n_2), and (s, n_3); and the last two are for the speed of the subject vehicle s and its distance to the lane closure point.

The two grid maps give the subject vehicle a global view of the current traffic conditions in the work zone, while the eight-element vector is to provide the subject vehicle with more detailed local traffic information. In total, the proposed RL method takes 3,208 state variables. Given such a large input dimension, it is reasonable to use neural networks to capture the learned control policy.

Neural Network Architecture. When the state space is discrete and compact, the Q function can be easily formulated as a table. When state space is continuous and multi-dimensional, however, it is impossible to formulate the Q function as a table or Monte Carlo tree (*23*) such as in AlphaGo Zero (*24*). In such a case, the Q function is often approximated by a parameterized function of states and actions $Q(s, a, w)$, and the learning process is to find the optimal parameter set w. This study adopts a modified impala convolutional neural network (*25*) to

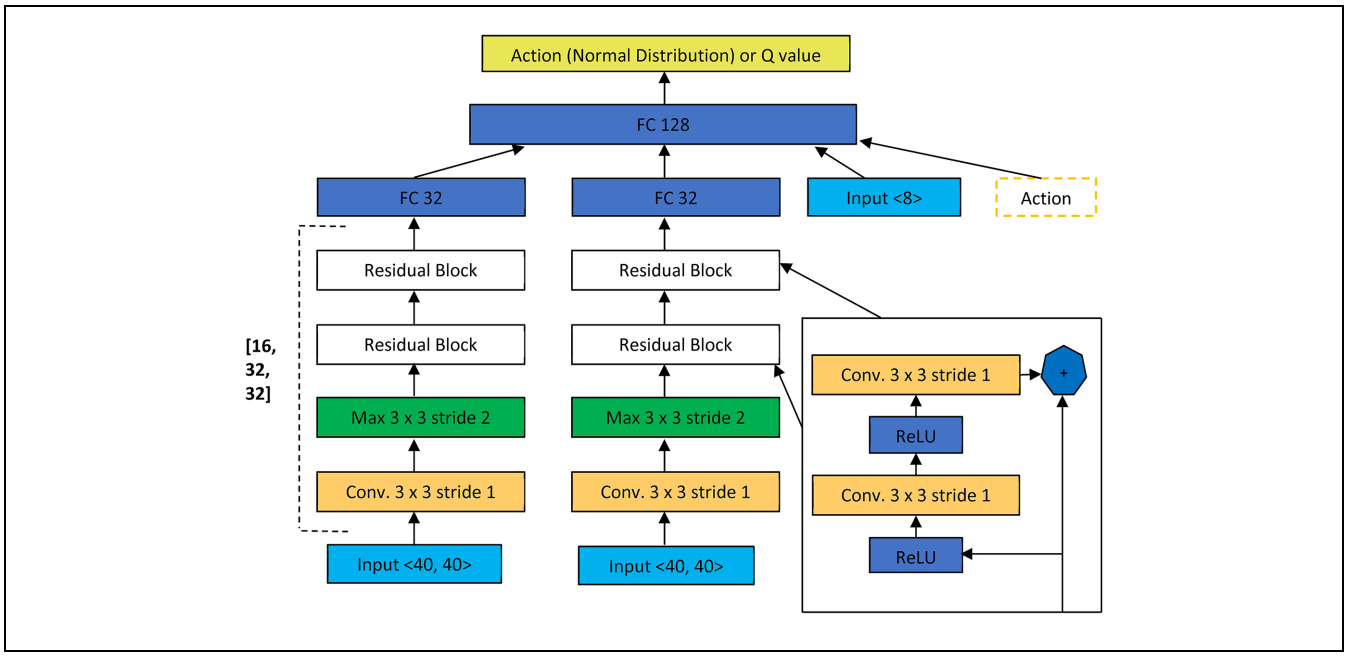

Figure 3. Convolutional neural network architecture.

approximate the Q function as well as the policy (actor) function.

As shown in Figure 3, the speed and acceleration grid maps are reshaped to two $<40, 40>$ matrices and fed into a convolutional neural network (CNN). The CNN includes three main blocks with filter sizes 16, 32, and 32, respectively. The first two main blocks correspond to the speed and acceleration grid maps, and the last block is for the eight-element vector. Each of the first two main blocks starts from a 3×3 convolutional layer, includes a 3×3 maxpooling layer down sampling with stride 2, and serves two residual blocks that have a similar architecture as ResNet (*26*). The reason for adopting this CNN architecture is that as the network depth increases, accuracy becomes saturated and degrades rapidly. The two residual blocks are included to increase the data sample efficiency by reusing activations from a previous layer until the adjacent layer learns its weights. This significantly simplifies the network and reduces the number of layers in it.

Via each of the first two main blocks, the raw state input is converted to a 32-dimension embedding that saves nonlinear and highly correlated information of the input. The two embeddings are concatenated with the eight-dimension vector as the input for the policy function, which output a final normal distribution with mean value and variance. The same CNN architecture is used for the Q function $Q(s, a)$ and value function $V(s)$. For the Q function, the action set is also added to the above concatenated vector to generate Q values for each state and action pair.

Soft Actor Critic. On-policy RL algorithms, such as proximal policy optimization (PPO) (*27*), asynchronous actor critic agents (A3C) (*28, 29*), and trust region policy optimization (TRPO) (*30*), although very popular, suffer from sample inefficiency because they need to generate new samples after each policy update and cannot utilize historical samples. On the contrary, Q-learning-based off-policy approaches, such as DDPG and DQN, are able to learn efficiently from past experience sampled from memory replay buffer. However, these off-policy optimization algorithms are very sensitive to hyperparameters and require a lot of tuning for the model to converge. To address this issue, this study uses a novel soft actor critic (SAC) RL. SAC is also an off-policy algorithm but includes new features to overcome the convergence brittleness problem.

SAC was initially proposed by Haarnoja et al. (*22*) and is adopted in this study. The main difference between SAC and other off-policy RL algorithms is that SAC aims to maximize both long-term rewards and the entropy of a policy. It encourages policy exploration by assigning approximately the same probabilities to actions that have similar Q values. These new features prevent the policy from always selecting a small set of actions with high Q values in the training process, while missing the chance of exploring other low Q-value actions that are potentially very rewarding in the long run. By encouraging policy exploration, SAC is able to address the convergence problem of other off-policy algorithms.

$$J(\theta) = \sum_{t=1}^{T} E_{(s_t, a_t) \sim \rho_{\pi_\theta}} [r(s_t, a_t) + \alpha \mathcal{H}(\pi_\theta(.|s_t))] \quad (1)$$

The policy function is obtained by maximizing the objective function in Equation 1, which consists of a reward term and an entropy term $\mathcal{H}$ weighted by α. SAC has three networks: a policy function π parameterized by Φ, a soft Q approximator function Q parameterized by θ, and a state value function V parameterized by ψ. The two separate approximators for V and Q functions are helpful for the learning process to converge.

To train the three CNNs, a series of loss functions are defined. The policy network π is trained by minimizing the following loss function in Equation 2:

$$\begin{aligned}\pi_{\text{new}} &= \arg\min_{\pi' \in \Pi} \mathrm{D_{KL}}\left(\pi'(\cdot|\mathrm{s_t})|\frac{\exp(\mathrm{Q}^{\pi_{\text{old}}}(\mathrm{s_t},.))}{\mathrm{Z}^{\pi_{\text{old}}}(\mathrm{s_t})}\right)\\ &= \arg\min_{\pi' \in \Pi} \mathrm{D_{KL}}(\pi'(.|\mathrm{s_t})|\exp(\mathrm{Q}^{\pi_{\text{old}}}(\mathrm{s_t},.) - \log \mathrm{Z}^{\pi_{\text{old}}}(\mathrm{s_t})))\end{aligned} \tag{2}$$

To update the policy network, SAC restricts the policy to a subset of policies Π which could be represented as a Gaussian distribution. In Equation 2, SAC uses the information projection defined in terms of the Kullback-Leibler divergence (*31*) between the old policy distribution and exponential of the old Q approximator function divided by the partition function Z which normalizes the old Q distribution. Function Z can be dropped since it is intractable in general and it does not affect the gradient with respect to the new policy.

Based on the Bellman equation, the soft Q-value can be computed iteratively starting from any function $Q : S \times A \rightarrow R$ given by

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma E_{s_{t+1} \sim \rho_\pi(s)}[V(s_{t+1})] \tag{3}$$

where

$$V(s_t) = E_{a_t \sim \pi}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)] \tag{4}$$

$V(S_t)$ in Equation 4 is the soft state value function. The soft state value function is trained by minimizing the squared residual error in

$$\mathrm{J_V}(\psi) = \mathrm{E}_{\mathrm{s_t} \sim \mathbb{D}}\left[\frac{1}{2}\left(\mathrm{V}_\psi(\mathrm{s_t}) - \mathrm{E}[\mathrm{Q}_\theta(\mathrm{s_t}, \mathrm{a_t}) - \log \pi_\Phi(\mathrm{a_t}|\mathrm{s_t})]\right)^2\right] \tag{5}$$

with gradient

$$\nabla_\psi \mathrm{J_V}(\psi) = \nabla_\psi \mathrm{V}_\psi(\mathrm{s_t})\left(\mathrm{V}_\psi(\mathrm{s_t}) - \mathrm{Q_w}(\mathrm{s_t}, \mathrm{a_t}) + \log \pi_\theta(\mathrm{a_t}|\mathrm{s_t})\right) \tag{6}$$

where D is the distribution of previously sampled states and actions saved in the replay buffer. The soft Q function is trained by minimizing the soft Bellman residual using the stochastic gradient descent method:

$$\mathrm{J_Q}(\theta) = \mathrm{E}_{(\mathrm{s_t}, \mathrm{a_t}) \sim \mathbb{D}}\left[\frac{1}{2}\left(\mathrm{Q}_\theta(\mathrm{s_t}, \mathrm{a_t}) - \left(\mathrm{r}(\mathrm{s_t}, \mathrm{a_t}) + \gamma \mathrm{E}_{\mathrm{s_{t+1}} \sim \rho_\pi(\mathrm{s})}\left[\mathrm{V}_{\bar{\psi}}(\mathrm{s_{t+1}})\right]\right)\right)^2\right] \tag{7}$$

with gradient

$$\begin{aligned}&\nabla_\mathrm{w} \mathrm{J_Q}(\mathrm{w})\\ &= \nabla_\mathrm{w} \mathrm{Q_w}(\mathrm{s_t}, \mathrm{a_t})\left(\mathrm{Q_w}(\mathrm{s_t}, \mathrm{a_t}) - \mathrm{r}(\mathrm{s_t}, \mathrm{a_t}) - \gamma \mathrm{V}_{\bar{\psi}}(\mathrm{s_{t+1}})\right)\end{aligned} \tag{8}$$

The target state value network $V_{\bar{\psi}}$ weights are updated by an exponential moving average considering the current value state network weights.

Reward Shaping. The main goal of reward shaping is to avoid creating "stop-and-go" traffic when a vehicle merges from the closed lane into the open lane. It requires the subject vehicle to keep a minimum safe distance with its lead vehicle and lag vehicle in the open lane when making a lane change. When the subject vehicle merges into the open lane, all vehicles surrounding it are supposed to continue smoothly without having to accelerate or decelerate.

Vehicles in the closed lane trying to merge are either in a non-terminal state or the terminal state. Non-terminal state represents when a vehicle is in Metering Zone II and adjusting its position, while terminal state is when a vehicle successfully merges into the open lane. The terminal state reward is calculated by

$$R = -\max(0, (70 - a_v)*0.2) \tag{9}$$

where a_v is the speed of the subject vehicle. The reward is negative if the subject vehicle is slower than 70 km/h, since this may create a backward shockwave. In addition, if $dx_1 > v_1 * th_{\min}$, $dx_2 > v_2 * th_{\min}$, and $|(v_s - (v_1 + v_2)/2)| < 2$, $R+ = 10$, where $th_{\min}$ is the minimum time headway, v is for speed, dx_1 is the distance headway between the subject vehicle and the lag vehicle in the target lane, and dx_2 is the distance headway between the subject vehicle and the lead vehicle in the target lane.

For non-terminal states, the reward is determined based on the following equations:

$$R = -0.01*\mathrm{acc}^2 \tag{10}$$

$$R -= 10 \text{ if a crash occurs} \tag{11}$$

$$\begin{aligned}&R -= 2.5 \text{ if } v_s < \|30 \text{ km/h}\| \; v_s > \\ &\|100 \text{ km/h}\| \text{ front headway} < 2\text{m}\end{aligned} \tag{12}$$

$$R -= (v_s - 80)*0.01 \text{ if } v_s > 80 \text{ km/h} \tag{13}$$

$$R -= (60 - v_s)*0.01 \text{ if } v_s < 60 \text{ km/h} \tag{14}$$

Equation 10 encourages the subject vehicle to drive smoothly with minimum acceleration/deceleration. Equation 11 means that the simulation will be

terminated and restarted if a crash occurs. Equations 12, 13, and 14 aim to minimize the vehicle's speed fluctuations around the speed limit (assuming 70 km/h in this study). The reward functions are carefully designed and help the subject vehicle learn how to follow the lead vehicle without a crash, travel at a reasonable speed, and maintain a safe distance with both the lead and lag vehicles in the target lane.

Simulation Analysis

Experiment Design

This research adopts VISSIM to evaluate the performance of the proposed RL control strategy and to compare it with EM, LM, CACC, and no control (Base case) under two input traffic volumes: 1,600 vehicles per hour (vph) and 2,000 vph. In the Base case, vehicles are completely controlled by the default VISSIM car-following and lane-changing models with a modified time headway, which is set to be 1.7 s based on the findings in Yang et al. (*32*). EM and LM are the same as the Base case except that vehicles are advised to change to the open lane at different locations. Also, EM and LM simply advise vehicles to change lanes, and the lane-changing maneuvers (e.g., adjust longitudinal position to find a suitable gap and change lane) are still carried out by the default VISSIM models. No metering zones or vehicle cooperation are considered in EM, LM, and the no control Base case.

Compared with the Base case, the CACC case has the following rule changes: (a) it assumes that the longitudinal controls of all vehicles are automated and no perception–reaction time is needed; and (b) these vehicles can form platoons in the metering zones. Similar to the Base case but different from EM and LM, in the CACC case VISSIM decides where a vehicle should change lanes. This is also different from the RL case, where vehicles are not allowed to change lanes in the metering zones.

RL also adds two changes to the Base case in the metering zones: (a) vehicles in the left (open) lane are still controlled by the default VISSIM model with a modified time headway (i.e., doubled) to create large gaps. The default perception–reaction time is still considered for these vehicles; and (b) vehicles in the right lane (i.e., to be closed) are controlled by the RL algorithm without considering perception–reaction time. They behave cooperatively and try to maintain appropriate gaps with surrounding vehicles to facilitate the merge downstream. Once entering the merging zone, vehicles in the right lane are required to change to the left open lane. The lane-changing process again is controlled by the default VISSIM models just like in the Base case.

As shown in Figure 1, a work zone on a two-lane highway with the right lane closed is considered. For all simulations conducted, the percentage of heavy vehicles is set as 3%, and the speed limit is set as 70 km/h. For each merge control and input volume combination, the simulation is run 10 times with different random seeds. Each simulation run lasts 45 min, with the first 15 min serving as the warm-up period.

Overall Mobility Performance

Table 1 shows the mobility performance for different control strategies. The typical capacity for a two-lane highway with one lane closed is about 1,340 vph (*33*). When the input volume is 1,600 vph (i.e., above the normal capacity), the RL control gives the best results for all performance measures, followed by CACC which generates very competitive results as well. Compared with EM and LM, the delay from RL control in this case is much smaller. The throughput generated by RL control is almost the same as the input, demonstrating its superior mobility performance. Not surprisingly, no control yields the worst results. The average throughput without any control is 1,343 vph, which is consistent with the capacity reported in (*33*).

When the input volume increases from 1,600 vph to 2,000 vph, even the average delay for RL control goes up significantly. However, the trend observed under the 1,600 vph input volume level still holds. For EM and LM, the percentage improvements in terms of average delay and mean travel time both drop significantly compared with the 1,600 vph demand level, while the percentage improvements in terms of throughput stay approximately the same. Compared with EM, LM, and the Base case, the overall performance of CACC is still much better, suggesting that CACC has great potential to improve highway work zone operations under medium to high traffic. Even with such a strong competitor, RL generates 83% less average delay and 50% higher mean travel time than CACC.

Overall, the results in Table 1 suggest that RL control significantly improves work zone mobility compared with both CACC and traditional control strategies like EM and LM. Under oversaturated condition (e.g., 2,000 vph), the performance differences between EM and LM become marginal, especially in terms of average delay and mean travel time. On the other hand, RL control performs the best under both congested and oversaturated conditions.

Vehicle Trajectory Diagram

To illustrate how RL control adjusts the positions of individual vehicles and the benefits of doing so, the

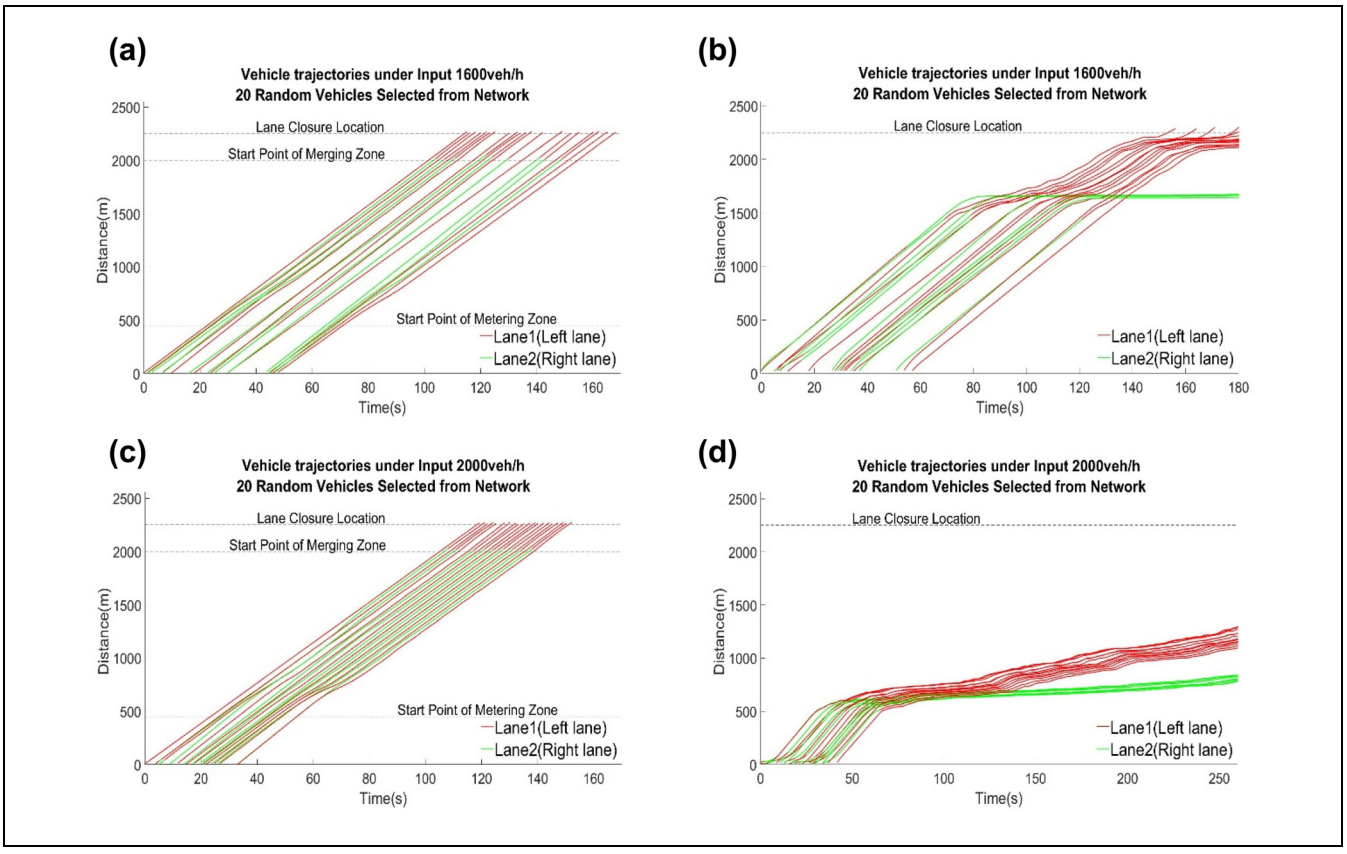

Figure 4. Vehicle trajectory diagrams: (*a*) reinforcement learning (RL) control under 1,600 vph, (*b*) no control under 1,600 vph, (*c*) RL control under 2,000 vph, (*d*) no control under 2,000 vph.

trajectories of vehicles in a randomly selected time frame are plotted in Figure 4. Each line represents the trajectory of a vehicle. The line color changes depending on which lane the vehicle is in. When the vehicle is in the right lane (to be closed), the trajectory line is green. When the vehicle changes into the left (open) lane, the trajectory line turns red. Ideally, all trajectory lines turn red before the lane closure point, meaning all vehicles are in the left (open) lane. Under RL control all green lines eventually turn red in the merging zone. For no control, however, Figure 4 clearly shows that many vehicles in the closed lane have to stop and wait for an extended period of time before they can merge into the open lane.

Other than the mobility benefits of RL control clearly illustrated in Figure 4, the slopes of the trajectories show that RL control can help reduce rear-end crash risk, by avoiding sudden decelerations and stop-and-go traffic. Additionally, the no control trajectories show that some drivers in the closed lane have to wait for an extended amount of time to be able to merge and may become increasingly impatient. This intuitively may contribute to aggressive and unsafe behaviors such as forced merge, and increase the risk of angle crashes.

Density

To investigate further how RL control performs, a VISSIM tool is developed to visualize how traffic density in the work zone changes over time and distance. The density maps for input volume = 1,600 vph under the RL control and LM strategies are presented in Figure 5, where the vertical axis is for time and the horizontal axis is for distance. A distance of 0 refers to the point 400 m upstream of the metering zone. Larger distance values are for locations downstream of the origin. Also, red colors are for higher densities. Figure 5 clearly shows that, compared with LM, the RL control can better reduce and equalize the traffic densities of the open and closed lanes. Equal densities in both lanes can help reduce drivers' desire for lane changes (e.g., seeking higher speeds) and consequently reduce angle crash risk. A smaller high-density area for RL control means the total vehicle time spent in stop-and-go traffic is less, suggesting that RL control is safer than LM at both input traffic volumes. Figure 5 also shows that the queues from the RL control grow at a much slower speed (i.e., backward forming shockwave speed) than the LM control. A slowly growing backward forming shockwave is likely to be less dangerous than a fast growing one.

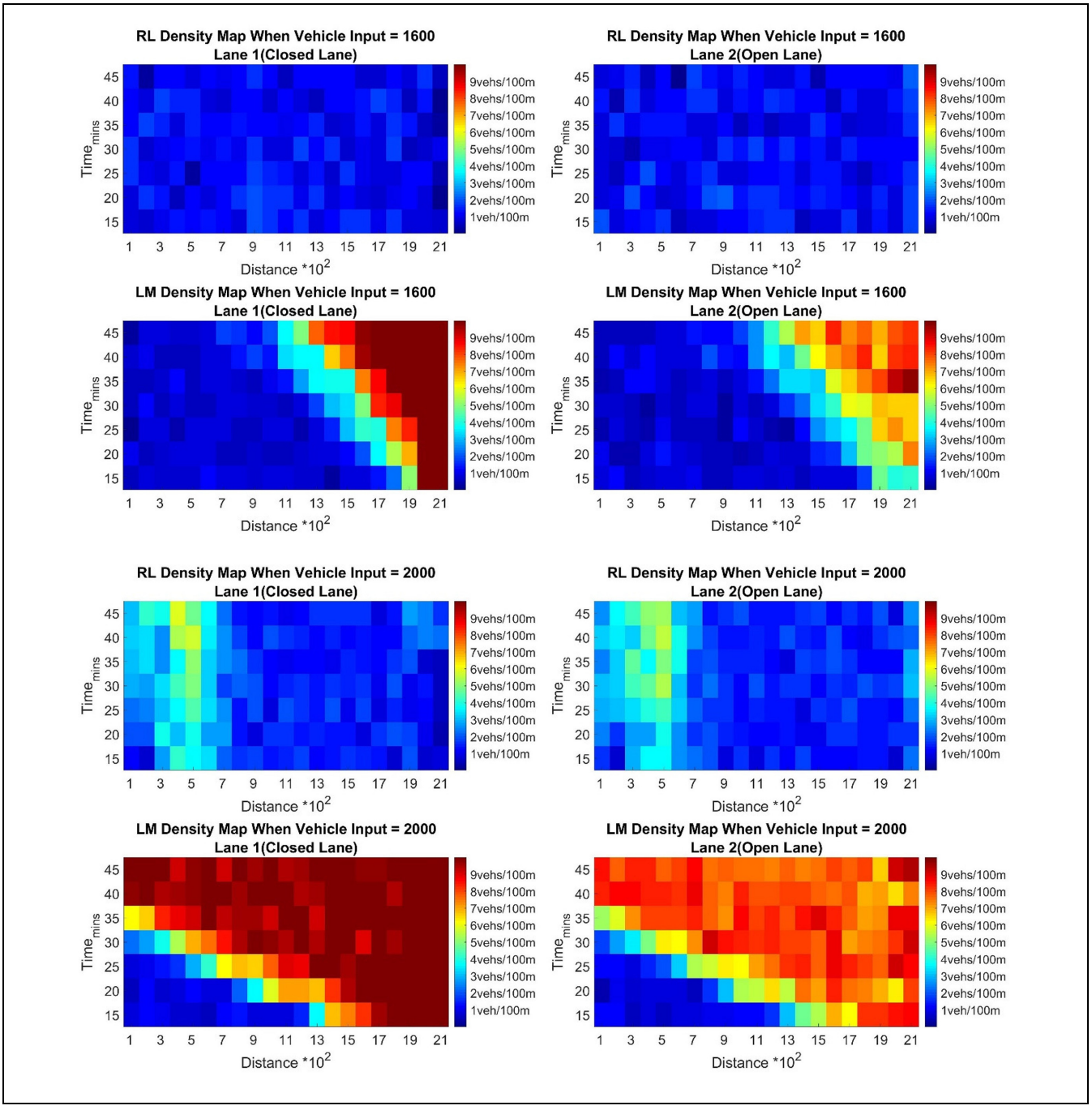

Figure 5. Reinforcement learning (RL) control and late merge (LM) density map comparison.

Acceleration and Distance Headway

The majority of crashes in highway work zones are rear-end crashes, which are often caused by sudden decelerations and stop-and-go traffic. Therefore, the stability of vehicle longitudinal acceleration behavior can be an important surrogate safety measure. Figure 6 shows the longitudinal acceleration distributions of vehicles under RL control and no control. The acceleration distributions for no control clearly are more spread out than those for RL control, and RL control generates much less sudden decelerations (e.g., $<= -5\,m^2/s$). This suggests that RL control is safer than no control and leads to smoother and more stable traffic flow.

The distance headway distributions for RL control and no control are also compared. Under both input flow conditions, overall RL control results in larger (safer) distance headways than no control in the merging zone.

Table 1. Performance of Different Control Strategies Compared

Performance measure	Merge control strategy				
	Base case	Early merge	Late merge	Cooperative adaptive cruise control	Reinforcement learning control
Volume input 1,600 vph					
Average delay (s)	274.8	121.9 (–55%)	64.8 (–76%)	15.7 (–94%)	4.2 (–97%)
Throughput (vph)	1343	1424 (6%)	1517 (13%)	1552 (15%)	1596 (19%)
Mean travel time (s)	384.3	231.4 (–40%)	174.3 (–55%)	129.7 (–66%)	116.5 (–70%)
Volume input 2,000 vph					
Average delay (s)	561.6	374.6 (–33%)	372.5 (–34%)	167.0 (–70%)	28.4 (–94%)
Throughput (vph)	1341	1436 (7%)	1526 (14%)	1686 (26%)	1979 (48%)
Mean travel time (s)	671.1	484.0 (–28%)	482.0 (–28%)	281.5 (–58%)	140.8 (–79%)

Note: Numbers in parenthesis are relative differences, which are calculated as (control case – base case)/(base case) × 100%. vph = vehicles per hour.

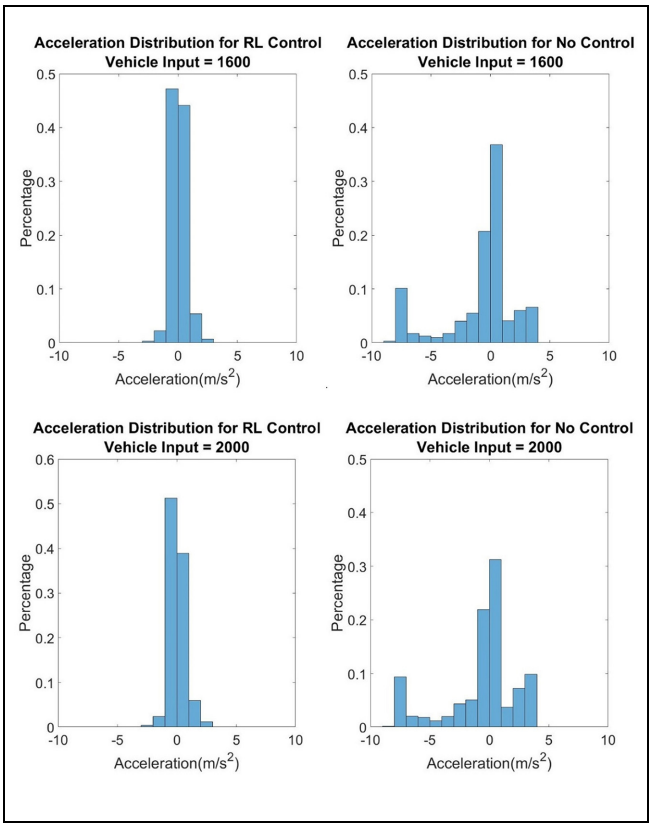

Figure 6. Acceleration histograms for reinforcement learning (RL) control and no control.

Discussion and Conclusion

This study proposes a cooperative highway work zone merge control strategy based on SAC RL. This strategy is evaluated using VISSIM simulation and compared with merging under conditions of: no control, LM, EM, and CACC merge. The RL-based control performs significantly better than the remaining control strategies under congested to extremely heavy traffic conditions in terms of both safety and mobility measures. Unlike other autonomous and connected vehicle control algorithms like CACC, which increases the capacity of the work zone by reducing vehicle time headway and/or reaction time, this RL-based control introduces two metering zones where vehicles adjust their positions relative to neighboring vehicles in the adjacent lane to achieve a collaborative and smooth merge and to maintain a safe time headway in the merging zone. The results also suggest the importance of AVs to collaborating with each other in order to improve the overall system operations.

The proposed RL-based control strategy is applied to a two-lane highway work zone example. It can be further modified for multi-lane (more than two-lane) highway work zones. For future studies, it would be interesting to investigate how to improve the system so that it can work

in an environment with both automated and human-driven vehicles. Also, further research can be done to optimize the lengths of the metering zones for improved system performance.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: Tianzhu Ren, Yuanchang Xie; data collection: Tianzhu Ren; analysis and interpretation of results: Tianzhu Ren, Yuanchang Xie, Liming Jiang; draft manuscript preparation: Tianzhu Ren, Yuanchang Xie, Liming Jiang. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The New England Transportation Consortium (NETC 14-4) provided financial support to this research. This work is also partially supported by the National Science Foundation (NSF # 1734521).

References

1. Tarko, A. P., S. R. Kanipakapatnam, and J. S. Wasson. *Modeling and Optimization of the Indiana Lane Merge Control System on Approaches to Freeway Work Zones, Part I*. Publication FHWA/IN/JTRP-97/12-1. Joint Transportation Research Program, Indiana Department of Transportation and Purdue University, West Lafayette, IN, 1998, p. 345.
2. Beacher, A. G., M. D. Fontaine, and N. J. Garber. *Evaluation of the Late Merge Work Zone Traffic Control Strategy*. Publication FHWA/VTRC 05-R6. Virginia Transportation Research Council, Virginia Department of Transportation, Federal Highway Administration, Washington, D.C., 2004.
3. Marsden, G., M. McDonald, and M. Brackstone. Towards an Understanding of Adaptive Cruise Control. *Transportation Research Part C: Emerging Technologies*, Vol. 91, 2001, pp. 33–51.
4. Chen, X., M. Jin, C. Chan, Y. Mao, and W. Gong. Bionic Decision-Making Analysis during Urban Expressway Ramp Merge for Autonomous Vehicle. Presented at 96th Annual Meeting of the Transportation Research Board, Washington, D.C., 2017.
5. Urmson, C., J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, and M. Grittleman. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, Vol. 25, No. 8, 2008, pp. 425–466.
6. Van Arem, B., C. J. Van Driel, and R. Visser. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 4, 2006, pp. 429–436.
7. Bojarski, M., D. Del Tesla, D. Zhang, B. Dworakowski, B. Firner, P. Flepp, L. D. Goyal, U. Monfort, J. Muller, and X. Zhang. End to End Learning for Self-Driving Cars. *arXiv Preprint arXiv:1604.07316*, 2016.
8. Endsley, M. R. Autonomous Driving Systems: A Preliminary Naturalistic Study of the Tesla Model S. *Journal of Cognitive Engineering and Decision Making*, Vol. 11, No. 3, 2017, pp. 225–238.
9. Desjardins, C., and B. Chaib-Draa. Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, 2011, pp. 1248–1260.
10. Chu, T., and U. Kalabić. Model-Based Deep Reinforcement Learning for CACC in Mixed-Autonomy Vehicle Platoon. *Proc., 58th IEEE Conference on Decision and Control*, Nice, France. IEEE, New York, 2019.
11. Wei, S., Y. Zou, T. Zhang, X. Zhang, and W. Wang. Design and Experimental Validation of a Cooperative Adaptive Cruise Control System Based on Supervised Reinforcement Learning. *Applied Sciences*, Vol. 8, No. 7, 2018, p. 1014.
12. Wang, P., and C. Y. Chan. Formulation of Deep Reinforcement Learning Architecture Toward Autonomous Driving for On-Ramp Merge. *Proc., 20th International Conference on Intelligent Transportation Systems*, Yokohama, Japan, IEEE, New York, 2017.
13. Bouton, M., A. Nakhaei, K. Fujimura, and M. J. Kochenderfer. Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic. *2019 IEEE Intelligent Transportation Systems Conference*, Auckland, New Zealand, IEEE, New York, 2019, pp. 3441–3447.
14. Fares, A., and W. Gomaa. Freeway Ramp-Metering Control Based on Reinforcement Learning. *Proc., 11th IEEE International Conference on Control & Automation*, Taiwan, 2014.
15. Yang, H., and H. Rakha. Reinforcement Learning Ramp Metering Control for Weaving Sections in a Connected Vehicle Environment. Presented at 96th Annual Meeting of the Transportation Research Board, Washington, D.C., 2017.
16. Howard, R. A. *Dynamic Programming and Markov Processes*. Technology Press of the Massachusetts Institute of Technology, Cambridge, MA, 1960.
17. Watkins, C. J., and P. Dayan. Q-Learning. *Machine Learning*, Vol. 8, No. 3–4, 1992, pp. 279–292.
18. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen. Human-Level Control Through Deep Reinforcement Learning. *Nature*, Vol. 518, No. 7540, 2015, pp. 529–533.
19. Yu, A., R. Palefsky-Smith, and R. Bedi. Deep Reinforcement Learning for Simulated Autonomous Vehicle Control. *Course Project Reports: Winter*, Stanford University, Stanford, CA, 2016, pp. 1–7.
20. Sallab, A., M. Abdou, E. Perot, and S. Yogamani. End-to-End Deep Reinforcement Learning for Lane Keeping

Assistance. *Proc., 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.
21. Lillicrap, P. T., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous Control with Deep Reinforcement Learning. *arXiv Preprint arXiv:1509.02971*, 2015.
22. Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv Preprint arXiv:1801.01290*, 2018.
23. Browne, C. B., E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, 2012, pp. 1–43.
24. Silver, D., J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, and Y. Chen. Mastering the Game of Go Without Human Knowledge. *Nature*, Vol. 550, No. 7676, 2017, pp. 354–359.
25. Krizhevsky, A., I. Sutskever, and G. E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, Vol. 60, 2012, pp. 84–90.
26. Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning. *Proc., 31st AAAI Conference on Artificial Intelligence*, San Francisco, CA, 2017.
27. Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv Preprint arXiv:1707.06347*, 2017.
28. Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *Proc., 33rd International Conference on Machine Learning*, New York City, NY, 2016, pp. 1928–1937.
29. Schulman, J., P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-Dimensional Continuous Control using Generalized Advantage Estimation. *arXiv Preprint arXiv: 1506.02438*, 2015.
30. Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust Region Policy Optimization. *Proc., 37th International Conference on Machine Learning*, Lille, France, 2015.
31. Hershey, J. R., and A. O. Peder. Approximating the Kullback Leibler Divergence between Gaussian Mixture Models. *Proc., 2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, Hawaii. IEEE, New York, Vol. 4, 2007.
32. Yang, N., G. L. Chang, and K. P. Kang. Simulation-Based Study on a Lane-Based Signal System for Merge Control at Freeway Work Zones. *Journal of Transportation Engineering*, Vol. 135, No. 1, 2009, pp. 9–17.
33. Dudek, C. L. Notes on Work Zone Capacity and Level of Service. Texas Transportation Institute, Texas A&M University, College Station, TX, 1984.

The data analysis, findings, and conclusions in this paper are the responsibility of the authors only and do not represent the official policy or opinion of the New England Transportation Consortium or the National Science Foundation.