Providing real-time en-route suggestions to CAVs for congestion mitigation: A two-way deep reinforcement learning approach

Xiaoyu Ma and Xiaozheng (Sean) He*

Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute,
110 8th St, Troy, NY 12180, United States

Keywords:
Information provision
Correlated Equilibrium
Reinforcement learning
Connected autonomous vehicles
Congestion mitigation

This research investigates the effectiveness of information provision for congestion reduction in Connected Autonomous Vehicle (CAV) systems. The inherent advantages of CAVs, such as vehicle-to-everything communication, advanced vehicle autonomy, and reduced human involvement, make them conducive to achieving Correlated Equilibrium (CE). Leveraging these advantages, this research proposes a reinforcement learning framework involving CAVs and an information provider, where CAVs conduct real-time learning to minimize their individual travel time, while the information provider offers real-time route suggestions aiming to minimize the system's total travel time. The en-route routing problem of the CAVs is formulated as a Markov game and the information provision problem is formulated as a single-agent Markov decision process. Then, this research develops a customized two-way deep reinforcement learning approach to solve the interrelated problems, accounting for their unique characteristics. Moreover, CE has been formulated within the proposed framework. Theoretical analysis rigorously proves the realization of CE and that the proposed framework can effectively mitigate congestion without compromising individual user optimality. Numerical results demonstrate the effectiveness of this approach. Our research contributes to the advancement of congestion reduction strategies in CAV systems with the mitigation of the conflict between system-level and individual-level goals using CE as a theoretical foundation. The results highlight the potential of information provision in fostering coordination and correlation among CAVs, thereby enhancing traffic efficiency and achieving system-level goals in smart transportation.

* Corresponding author. E-mail address: hex6@rpi.edu.

1 Introduction

Congestion reduction has been one of the core research topics in highway traffic management for a long time. Central to this topic is the conflict between user optimality (UO) and system optimality (SO). Under UO, travelers attempt to minimize their own travel disutility selfishly, often leading to increased traffic congestion. On the other side, traffic management authorities strive to deploy effective strategies to improve traffic efficiency, usually requiring travelers to deviate from UO to minimize the total system travel time, defined as SO. However, due to operational and policy-making challenges, achieving this goal is never trivial in practice for most theoretically efficient instruments.

One promising approach to resolve the UO-SO conflict is to foster correlation or coordination among travelers through the involvement of a third party. This allows the system to deviate from UO. Information provision is an example of such a third-party approach, which has been extensively analyzed and implemented in Advanced Traveler Information Systems (Ma et al., 2016; Paz and Peeta, 2009; Du et al., 2015; Spana et al., 2022). However, challenges arise when the trustworthiness of the third party is concerned if the travelers perceive that their UO is compromised. This highlights the importance of developing a third-party mechanism that can achieve system-level goals without compromising individual UO. In game theory, the concept of Correlated Equilibrium (CE) (Aumann, 1987) offers an opportunity to resolve the UO-SO conflict, where players can mutually obtain higher payoffs by following the guidance of a third party pursuing system-level goals (Marris et al., 2021).

In transportation literature, Correlated Equilibrium (CE) has been applied to routing problems, where an information provider acts as the third party (Gairing et al., 2008; Liu and Whinston, 2019; Ning and Du, 2023). However, existing studies are limited to static information provision on small networks due to the computational burden and poor analytical properties of the underlying optimization sub-problems, especially in dynamic contexts. For general networks, even the design of a static pre-trip routing strategy is proved to be non-convex and nonlinear (Ning and Du, 2023). In addition, behavioral issues such as traveler compliance, perception error, and inertia hinder the effectiveness of information provision in practice. Therefore, achieving CE in large-scale networks remains challenging, especially in a dynamic context of a real-world traffic system with human-driven vehicles.

This research aims to answer a natural question: Can information providers effectively foster the realization of CE in emerging smart transportation systems, leveraging the power of Connected Autonomous Vehicle (CAV) technology? CAV systems possess inherent advantages for achieving CE. First, the connected-vehicle technology enables information providers to promote correlated behaviors among CAVs through Vehicle-to-Infrastructure communications by disseminating properly designed travel information aligned with system-level goals. Second, advanced vehicle autonomy, supported by edge computing capabilities, allows vehicles to learn and update their travel strategies in real-time and in a distributed manner to pursue their individual goals. Third, since CAV systems involve less human actions, concerns regarding behavior-related issues can be mitigated. These advantages faciliate a smarter transportation system with coordinated behaviors among users, offering new opportunities for simultaneously achieving system-level and individual-level goals.

This research concentrates on congestion reduction in CAV systems by fostering CE. Our modeling approach considers CAVs and the information provider as reinforcement learning agents pursuing individual- and system-level goals, respectively. CAVs conduct en-route learning in real-time, aiming to minimize their travel time. The information provider offers real-time en-route routing suggestions to CAVs, aiming to minimize the total travel time of the system. The learning capabilities of the agents in CAV systems allow us to formulate a multi-agent Markov routing game for CAVs and a single-agent Markov decision process for the information provider. To capture the unique features of the problem and enhance learning performance, this research develops a customized two-way reinforcement learning approach for the CAVs and the information provider. Theoretical analysis of CE within the proposed framework provides insights and a foundation for mitigating UO-SO conflict. Numerical results also demonstrate reduction in total system travel time without compromising individual goals.

The contributions of this research are summarized as follows:

- Formulation of a Markov game for CAVs conducting en-route choices, considering the unique characteristics of the problem;
- Formulation of a single-agent Markov decision process for the information provider, providing en-route suggestions based on real-time traffic conditions;
- Development of a two-way deep reinforcement learning framework and algorithm for CAVs and the information provider, capturing their real-time interactions;
- Customization of reinforcement learning methods to accommodate the unique characteristics of the formulated problems and enhance learning performance;
- Formulation and theoretical analysis of CE within the proposed framework, with rigorous proof that the learning converges towards SO without compromising UO.

The remainder of the paper is structured as follows. Section 2 presents the related literature review and emphasizes the research gap. Section 3 first presents some preliminary, then formulates the multi-agent Markov game for CAVs and the Markov decision process for the information provider. Section 4 proposes the customized two-way deep reinforcement learning approach and algorithm. Section 5 introduces, formulates, and theoretically analyzes CE in the proposed framework. Section 6 conducts numerical examples. The paper is concluded by Section 7.

2 Literature Review

This section conducts the literature review in three domains relevant to our research focus: (1) Information provision in CAV systems; (2) Markov game and multi-agent reinforcement learning for vehicle routing; (3) Applications of CE in transportation.

In the literature related to information provision in CAV systems, one stream of research focuses on improving local traffic performance (Zhou et al., 2022; Liang et al., 2023) or safety (Kim et al., 2019; Jo et al., 2022; Ko et al., 2023). This branch of studies is less related to our research. Therefore, we omit the details of them. Another stream of research focuses on achieving network-wide goals, e.g., SO, by information provision. Among them, Wang et al. (2021) consider enhancing traffic efficiency for CAV systems by providing route suggestions. The routing strategy relies on monetary incentives to ensure user participation and requires behavior assumptions such as honesty. Du et al. (2015) propose a coordinated routing mechanism with intentional information provision perturbation to balance UO and SO, by exploiting bounded rationality of the users. Spana et al. (2022) integrate strategic real-time traffic information perturbation into a coordinated routing mechanism for CAVs using a mixed-strategy congestion game to mitigate congestion. Spana and Du (2022) solve the optimal information perturbation for traffic congestion mitigation, based on Gaussian process regression and optimization. These three studies, i.e., (Du et al., 2015; Spana et al., 2022; Spana and Du, 2022), present solid efforts in integrating, analyzing, and solving the information perturbation on travel time to mitigate congestion. However, the loss of UO still exists when pursuing SO, which can cause user dissatisfaction. In addition, none of the above research considers en-route choices and the dynamical features of the CAV system.

Markov game and multi-agent reinforcement learning have been applied in dynamic routing behavior for transportation systems. There are two main types of route decisions considered: the decision of a complete route choice and en-route decisions. In the research focusing on complete route choice behaviors, e.g., (Stefanello et al., 2016; de O. Ramos et al., 2018; Zhou et al., 2020), the action space for an agent is the route set from origin to destination. These studies assume that agents follow the chosen routes until they reach their destination without en-route changes. The problems are similar to using reinforcement learning to find the shortest routes without response to traffic dynamics. For the research focusing on the en-route context, two studies Grunitzki et al. (2014) and Bazzan and Grunitzki (2016) model the multi-agent en-route behaviors using reinforcement learning, with location as state and next-to-go link as action. However, these studies assume each agent is an independent learner and ignores the interaction among agents, which is not realistic and suffers from instability issues. Shou et al. (2022) formulate a Markov routing game for agents and solves it using mean field multi-agent reinforcement learning, capturing the competition among agents and congestion effects. However, the research does not explicitly factor the unique features of en-route behaviors in the modeling. For example, in their algorithm, agents take joint actions every time without factoring the different en-route statuses of different agents. In addition, Su et al. (2023) propose a multi-agent reinforcement learning framework for traffic signal control for emergency vehicle routing to reduce congestion. Our research formulates the en-route Markov game and designs the multi-agent reinforcement learning framework for CAVs, capturing the competition among agents, congestion effect, and the unique features of en-route behaviors. Additionally, we also integrate the learning of an information provider with the CAVs and propose a two-way reinforcement learning framework to achieve individual and system-level goals simultaneously.

The concept of CE receives little attention in transportation research. The topics include applying CE in traffic management with unmanned aerial vehicles (Tony et al., 2022), minimizing delay at intersections (Adkins et al., 2019), and mitigating network-level congestion through information provision (Liu and Whinston, 2019; Gairing et al., 2008; Ning and Du, 2023). Among the three studies in the third category that relates to this paper, Liu and Whinston (2019) and Gairing et al. (2008) use simple and small networks to study and analyze the CE with information provision. Specifically, Liu and Whinston (2019) propose an information design framework through Bayes correlated equilibrium to improve the traffic efficiency. Gairing et al. (2008) propose a Bayesian routing game for selfish users with incomplete information. The study proved the existence of CE in the routing game and analyzed the computation complexity on simple networks. Recently, Ning and Du (2023) propose a routing mechanism for traffic congestion mitigation built upon CE, and design robust algorithms to solve the optimal routing. Their work addresses the conflict between UO and SO by involving an information provider providing pre-trip route suggestions.

While sharing the research interests similar to Ning and Du (2023), our research presented in this paper focuses on a dynamic en-route context for the systems with self-learning CAVs and an information provider who is also learning the optimal information provision strategy. We intend to address the challenges in solving the UO-SO conflict and reduce the behavior constraints and implementation complexity of CE in practice by leveraging emerging technologies.

3 Model Formulation

This research considers a traffic system where the vehicles are intelligent CAVs that possess self-learning capabilities supported by advanced computation resources. Each vehicle behaves selfishly with the aim of minimizing its own travel time. While an information provider is a central agency with the objective of minimizing the total travel time of the system by disseminating travel information to each CAV.

Each CAV is an intelligent agent making individual en-route choices according to its observed traffic condition. Their actual travel times are affected by other CAVs' behaviors. Since each CAV behaves selfishly, there is competition for the shared resources (road capacity) among CAVs. And the resulting joint behaviors usually lead to compromise in system-wide efficiency, i.e., an increase in congestion. In this research, the routing problem of CAVs is formulated as a Markov Game (Littman, 1994), in which each agent behaves non-cooperatively, and their payoffs are influenced by each other. Each CAV conducts reinforcement learning to learn its optimal policy (en-route strategy), and this forms a Multi-Agent Reinforcement Learning (MARL) problem.

The information provider provides en-route suggestions to each CAV based on the real-time traffic condition to influence the CAVs' behaviors, with the objective of minimizing the total travel time of the system. After disseminating the suggestions to CAVs, in real time, the information provider can observe the aggregate travel behaviors of CAVs and the change in traffic conditions. From such observation, the information provider can adjust its strategy to minimize the total travel time of the system. This forms a learning process on the information provider's side. In this research, the information provision problem is formulated as a single-agent Markov Decision Process (MDP) (Puterman, 2008) that will be solved by reinforcement learning.

The shared environment for the CAVs and the information provider is the traffic condition in the system, which is dynamic as the traffic evolves. Essentially, both the CAV side and the information provider side learn their optimal policies by interacting with the environment, in which the behaviors of both sides influence each other. Therefore, a two-way reinforcement learning setting is established, which contains the CAV side and the information provider side. Figure 1 shows the conceptual framework. In the following, Section 3.1 introduces preliminaries of the definitions of single-agent MDP and multi-agent Markov game. Then Section 3.2 and Section 3.3 present the proposed formulations for the Markov game of CAVs and the MDP for the information provider, respectively.

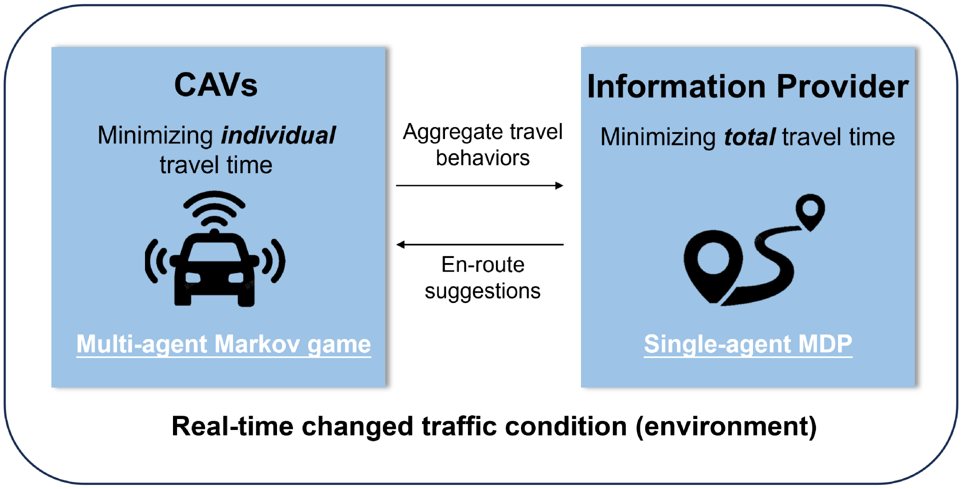

CAVs
Minimizing *individual* travel time
Multi-agent Markov game
Aggregate travel behaviors
En-route suggestions
Information Provider
Minimizing *total* travel time
Single-agent MDP
Real-time changed traffic condition (environment)

Figure 1: Conceptual framework

3.1 Preliminaries of MDP and Markov game

The Markov Decision Process (MDP) is concerned with a single agent learning a policy that optimizes a numerical objective by making sequential decisions. The agent interacts with an environment of unknown dynamics in a trial-and-error fashion and receives feedback after executing a decision (Gronauer and Diepold, 2022; Zhang et al., 2021). The standard formulation for such a sequential decision-making process, i.e., MDP, is defined as follows (Bellman, 1957; Puterman, 2008; Zhang et al., 2021).

Definition 1. *A Markov decision process is defined by a tuple (S, A, P, R, γ), where S and A denote the state and action spaces, respectively; $P : S \times A \to \Delta(S)$ denotes the transition probability from any state $s \in S$ to any state $s' \in S$ for any given action $a \in A$; $R : S \times A \times S \to \mathbb{R}$ is the reward function that determines the immediate reward received by the agent for a transition from (s,a) to s'; $\gamma \in [0,1]$ is the discount factor that trades off the instantaneous and future rewards.*

In an MDP, at each time t, the agent chooses an action $a_t \in A$ to execute, given any non-terminal state $s_t \in S$. After execution, it observes a new state $s_{t+1} \in S$ with probability $P(s_{t+1}|s_t, a_t)$ and receives a reward $r(s_t, a_t, s_{t+1}) \in \mathbb{R}$. The goal of solving the MDP is to find a policy $\pi : S \to \Delta(A)$, a mapping from the state space S to the distribution over the action space A, so that $a_t \sim \pi(\cdot|s_t)$ and the expected discounted accumulative reward (Equation (1)) is maximized,

$$\mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0\right] \tag{1}$$

where s_0 is the initial state, and γ is the discount factor. When $\gamma = 1$, the agent does not differentiate future rewards from immediate rewards. As γ gets smaller, the agent cares less about rewards received in the distant future.

Accordingly, the state-value function (V-function) and action-value function (Q-function) under policy π are defined as Equation (2) and Equation (3), respectively.

$$V_\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0 = s\right] \tag{2}$$

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), a_0 = a, s_0 = s\right] \tag{3}$$

For any state $s \in S$ and action $a \in A$, $V_\pi(s)$, $V_\pi(s)$ is the expected discounted accumulative reward starting from $s_0 = s$, and $Q_\pi(s,a)$ is the expected discounted accumulative reward starting from $(s_0, a_0) = (s, a)$.

An optimal policy, π^*, is the policy that maximizes the value function, i.e., $\pi^* = argmax_\pi V^\pi(s), \forall s \in S$. The functions $V^*_\pi(s)$ and $Q^*_\pi(s,a)$ corresponding to the optimal policy π^* are called the optimal value function and the optimal action-value function, respectively. The optimal value function, $V^*_\pi(s)$, can be written in a recursive way as shown in Equation (4), in relation to the next state s' given state s.

$$V^*(s) = \max_a \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[r(s,a,s') + \gamma V^*(s')\right] \tag{4}$$

Mathematically, $V^*(s) = \max_a Q^*(s,a)$, and the following holds.

$$Q^*(s,a) = \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[r(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right] \tag{5}$$

The optimal policy can be derived as $\pi^*(s) = argmax_a\, Q^*(s,a), \forall s \in S$.

Extended from MDP where there is only a single agent, Markov game describes the sequential decision process for multiple agents. In particular, the environment state and reward received by each agent are influenced by the joint actions of all agents. Each agent has its own cumulative reward to optimize, which now becomes a function of the policies of all other agents, in contrast to the single-agent case. The Markov game is defined as follows (Littman, 1994; Zhang et al., 2021).

Definition 2. *A Markov game is defined by a tuple $(N, S, \{A_i\}_{i\in N}, P, \{R_i\}_{i\in N}, \gamma)$, where $N = \{1, ..., n\}$ denotes the set of n agents, S denotes the state space observed by all agents, A_i denotes the action space of agent i. Let $A := A_1 \times ... \times A_n$, then $P : S \times A \to \Delta(S)$ denotes the transition probability from any state $s \in S$ to any state $s' \in S$ for any joint action $\boldsymbol{a} \in A$; $R_i : S \times A \times S \to \mathbb{R}$ is the reward function that determines the immediate reward received by agent i for a transition from $(s, \boldsymbol{a})$ to s'; $\gamma \in [0,1]$ is the discount factor.*

At time t, each agent $i \in N$ executes an action $a_{i,t}$, according to the environment state s_t. Then the environment transitions to the next state s_{t+1}, and gives rewards $r_i(s_t, \boldsymbol{a}_t, s_{t+1})$ to each agent i. The goal of each agent is to find a policy $\pi_i : S \to \Delta(A_i)$ that optimizes its own expected cumulative reward. In the multi-agent setting, the value function $V_i(s)$ and the action-value function $Q_i(s, \boldsymbol{a})$ become functions of the joint actions of all agents and the environment state.

The solution concept of a Markov game is different from that of an MDP, since the optimal performance of each agent is controlled not only by its own policy, but also by the policies of all other agents. There are several well-known solution concepts in Markov games, including Nash equilibrium (NE), correlated equilibrium (CE), and coarse correlated equilibrium (CCE) (Aumann, 1987; Chen et al., 2022; Mao and Başar, 2022).

In a more general case, either in an MDP or a Markov game, the state may not be fully observable to the agents, i.e., the agents can only observe a part of the state. In this case, the agents take actions based on their observation and establish the (action-) value functions based on observation instead of state. The policy becomes a mapping from the observation space to the action space.

3.2 Markov routing game for CAVs under information provision

This section presents the proposed Markov routing game for CAVs under information provision, where each CAV behaves non-cooperatively and tries to minimize its own individual travel time. The CAVs make en-route choices during the trip based on their latest observation of the traffic condition and received en-route suggestions from the information provider. In this research, the en-route choices are considered to be made at each intersection or interchange by selecting which subsequent link to travel. By constantly interacting with the environment, the CAVs accumulate trip experiences and learn their optimal en-route policies.

The Markov game for describing CAVs' traveling behaviors has distinct features that differ from the standard Markov game by definition; therefore, the formulation of the game should be customized to accommodate specific characteristics. Below presents three unique characteristics that must be taken into account in the formulation. In the rest of the paper, we use CAV and agent interchangeably.

- In the standard Markov game, agents take actions at the same time at each step and end the game together. However, for traffic systems, each CAV takes action only at the time when it arrives at an intersection or interchange, which differs from one CAV to another since different CAVs can have different trips and do not make decisions at the same time. And they can finish their trips (end the game) at different times.

- The change in environment state is defined to be triggered by the joint action of the agents at each step in a standard Markov game. While in the traffic system, the traffic environment changes right after any CAV takes an action instead of the joint action. This means that, from one CAV's perspective, the environment can change even though it has not taken any new action.

- The size of the action set of a CAV varies along the trip since the number of subsequent links that are available varies at different intersections/interchanges.

Explicitly incorporating the above characteristics, this research formulates the Markov game for CAVs making en-route choices. Consider a traffic network $G(N,L)$ consisting of a set of nodes, $n \in N$, and a set of directed links, $a \in L$. Denote W as the set of Origin-Destination (OD) pairs. $D_w, w \in W$ is the set of the decision nodes in OD pair $w \in W$, which are the nodes of intersections/interchanges where CAVs can make en-route decisions. Denote $\overline{P}_{n,w}$ and $A_{n,w}$ as the set of the sub-routes and the set of the actions (links) at the decision node n for the travel in OD pair $w \in W$. A sub-route in $\overline{P}_{n,w}$ is defined as a route from decision node n to the destination in OD pair w. Consider a period of time of interest, discretized into time steps. Denote the set of the time steps as T. Below presents the details of each component of the formulated Markov game.

Agent. Each CAV is an agent, denoted by $i \in \{1,2,...,I\}$. The CAV i makes en-route choices at each decision node $n \in D_w$ during its trip in OD pair $w \in W$.

State. Since it is a multi-agent problem, the state refers to the environment state. In this research, the environment state $s \in S$ is the traffic condition. Specifically, we use the travel time of each link to represent the state. The environment changes at each time step as traffic evolves and CAVs take actions.

Observation. Each CAV traveling in OD pair $w \in W$ has an observation of the environment state at a time, which is part of the entire environment that relates to its travel in $w \in W$. In this research, we consider the observation $o_i \in O_i$, $i = 1,2,...,I$, includes the current time $t \in T$, location (decision node) $n \in D_w$, real-time travel time information of each sub-routes $\bar{t}_{\overline{p}}$, $\overline{p} \in \overline{P}_{n,w}$ (can be obtained by any channels such as navigation applications), and the suggested sub-route $\overline{p} \in \overline{P}_{n,w}$ provided by the information provider. Based on the observation, the CAV takes action at each decision node in $w \in W$.

Action. At each decision node $n \in D_w$, based on its observation, the CAV takes an action. That is, the CAV chooses a subsequent link a to travel from its action set $A_{n,w}$, which contains all the available links it can take at the decision node $n \in D_w$. The action set varies by the decision node it arrives at.

Reward. After taking an action at a decision node at time step $t \in T$, the CAV travels on the subsequent links until it reaches the next decision node at time $t' \in T$. At the new decision node, the CAV will receive a reward $r \in \mathbb{R}$ that is associated with the travel time it took from the last decision node. In this research, the reward is considered to be the negative of such travel time, i.e., $-(t'-t) \times h$, where h is the length of each time step. Then the CAV gets a new observation and takes an action at the new decision node.

Episode. For each CAV, an episode means a complete trip. The objective of each CAV i is to maximize its expected discounted accumulative reward in an episode,

$$\mathbb{E}\left[\sum_t \gamma^t r(o_{i,t}, a_{i,t}, o_{i,t'}) \middle| a_{i,t} \sim \pi_i(\cdot|o_{i,t}), o_{i,0}\right]$$

where t denotes the time steps when the CAV i needs to take action and t' is the next action time after t. Since this is a multi-agent problem, the reward received by a CAV is influenced by the environment state s_t and the actions of other CAVs, in addition to its own action. However, the environment state and the actions of other CAVs are not fully observable to the agent. Therefore, in the above equation, we just use the observation and the action of its own in the reward function, in which the environment state and the actions of other CAVs are partially reflected in the observed travel times by the definition of o_i. This research considers $\gamma = 1$; that is, each CAV wants to maximize its total travel time in a trip without any discount on the reward received later in the trip. From the CAV's perspective, the episode ends when they finish their own trip. While from the perspective of the system, the episode ends when all CAVs arrive at their destinations.

Policy. As trip experience accumulates, each CAV i traveling in OD pair $w \in W$ tries to learn an optimal policy π_i^*, which is a mapping from its observation space O_i to its action space $A_i = \{A_{n,w}, n \in D_w, w \in W\}$.

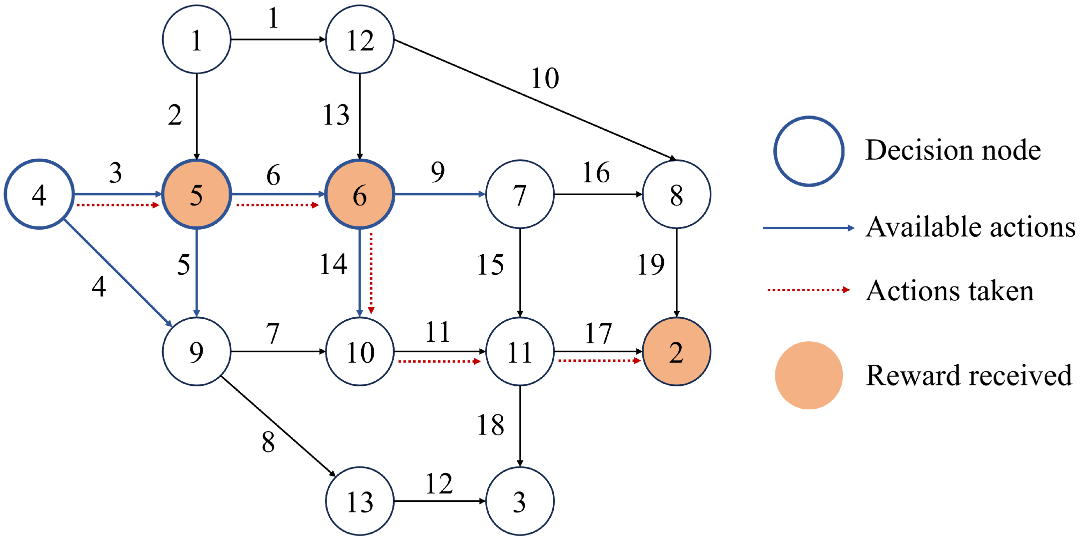


Figure 2: Illustration on Nguyen-Dupuis Network

Figure 2 illustrates the en-route process of one CAV in one episode (trip) on the Nguyen-Dupuis network. In this example, the CAV departs from the origin (node 4) toward the destination (node 2). At the origin, where is a decision node, the CAV chooses from link 3 and link 4 based on its current observation and policy. Assume it takes link 3, then it arrives at node 5 and receives a reward, which is the negative travel time it took from node 4 to node 5. Then the CAV chooses from link 5 and link 6 based on its current observation and policy. Assume it takes link 6, and then arrives at node 6. Similarly, it receives a reward that is the negative travel time it took from node 5 to node 6. Then it renews observation and takes action again. Now assume it takes link 14, then it will arrive at node 10, which is not a decision node. Note that the CAV now has only one sub-route option, which is link 11 $\rightarrow$ link 17. Hence, the CAV continues to travel through link 11 and link 17, and finally arrives at the destination (node 2). Then the CAV receives a reward, which is the negative travel time it took from the last decision node (node 6) to the current location (node 2). The total reward received in one episode is exactly the negative of the total travel time it took during the trip. Each CAV repeats such a process and accumulates experience, from which they learn their optimal policies.

3.3 MDP for the information provider

The information provider provides en-route suggestions to CAVs, i.e., a suggested sub-route, when the CAVs take actions at a decision node during the trip. The objective of the information provider is to minimize the system's total travel time, differing from the objective of individual CAVs. As aforementioned, the sequential decision process of the information provider, in terms of how to provide the suggested sub-routes, is formulated as an MDP to be solved by reinforcement learning. Below presents the details of each component of the MDP.

Agent. There is one single agent, which is the information provider.

Episode. The episode for the information provider starts from the beginning of the time period of interest and lasts until all the considered CAVs arrive at their destinations, e.g., a day, or the peak hours.

State. The state of the information provider is the travel time on each link in the system. It shares the same environment as the CAVs. However, when the information provider provides the en-route suggestion for a CAV, it only refers to the travel times relevant to the specific trip of the CAV. In this research, the information provider provides suggestions based on the current travel times of the sub-routes at the specific decision node instead of the entire system in which the traffic condition very far away has little impact.

Action. The information provider takes actions at each time step $t \in T$. Specifically, the action of the information provider refers to the probability distributions of suggesting each sub-route available at the decision nodes in all OD pairs in the system. Denote $\widehat{\boldsymbol{a}}_{n,w} \in \mathbb{R}^k$ as the action at the decision node n in OD pair w, which is a $k_{n,w}$ dimensional vector with each element as the probability of suggesting one corresponding sub-route, where $k_{n,w}$ is the number of sub-routes at the decision node $n \in D_w$. Therefore, the action of the information provider is indeed continuous. When a CAV arrives at a decision node $n \in D_w$, the information provider provides a suggested sub-route according to the probability distribution $\widehat{\boldsymbol{a}}_{n,w} \in \mathbb{R}^k$. From the perspective of the CAV, this suggested sub-route, as previously presented, is part of its observation. In the learning, to reduce the dimension of the problem, the action $\widehat{\boldsymbol{a}}_{n,w} \in \mathbb{R}^k$ can be represented as the probability distribution among the links available at a decision node since multiple sub-routes can share the same link.

Reward. After the information provider takes the action $\{\widehat{\boldsymbol{a}}_{n,w}, n \in D_w, w \in W\}$ at time t, it will receive a reward in the next time step t'. The reward is associated with the total travel time taken by all the CAVs in the previous time step. In this research, we use the negative total travel time of all the CAVs, i.e., $-(t'-t) \times h \times I_t$, as the reward received by the information provider at t', where I_t is the number of agents that travels in time step t. Therefore, the total reward received by the information provider in one episode is the negative total travel time of all CAVs.

Policy. The information provider seeks to learn a policy that determines its action $\widehat{\boldsymbol{a}}_{n,w} \in \mathbb{R}^k, n \in D_w, w \in W$ based on its state $s \in S$. The objective is to find a policy that minimizes the expected discounted accumulative reward,

$$\mathbb{E}\left[\sum_t \gamma^t r(s_t, [\widehat{\boldsymbol{a}}_{n,w,t}, n \in D_w, w \in W], s_{t'}) \middle| \widehat{\boldsymbol{a}}_{n,w,t} \sim \widehat{\pi}(\cdot|s_t), s_0\right]$$

where we also let the discount factor $\gamma = 1$ for the information provider. For a training task, a value close to 1 may be used for γ, such as 0.99, to help with convergence, according to the common practice rule in reinforcement learning.

So far, we have formulated the Markov game for CAVs and the MDP for the information provider. The shared environment is indeed model-free traffic dynamics. Next, we present the approach for solving the problems.

4 Customized Two-Way Reinforcement Learning Approach

Reinforcement learning (RL) is a typical technique used for solving model-free MDPs and Markov games. In this section, we propose a two-way deep RL framework, in which we have the multi-agent RL for CAVs and the single-agent RL for the information provider. Figure 3 shows the main components of the framework.

In Figure 3, the CAVs and the information provider share the same environment of the traffic system. The CAVs take actions at each decision node, while the information provider takes action at each time step. To account for the characteristics of the problems, we apply deep Q-learning to the Markov game of CAVs and use an actor-critic method for the information provider. The subsequent sub-sections will provide detailed explanations of the RL processes.

In this framework, the CAVs obtain observations from the environment and the information provider, and take actions that will impact the environment. The information provider obtains state from the environment and takes action that will impact the CAVs behaviors, and in turn, impact the environment. The information provider and the CAVs interact with other and conduct RL at the same time. In fact, the information provider repeatedly learns the aggregate behaviors of CAVs, from which it learns the strategy of providing suggestions that impact the actions of CAVs. The CAVs also interact with the environment and the information provider to learn the optimal policy for taking en-route actions.

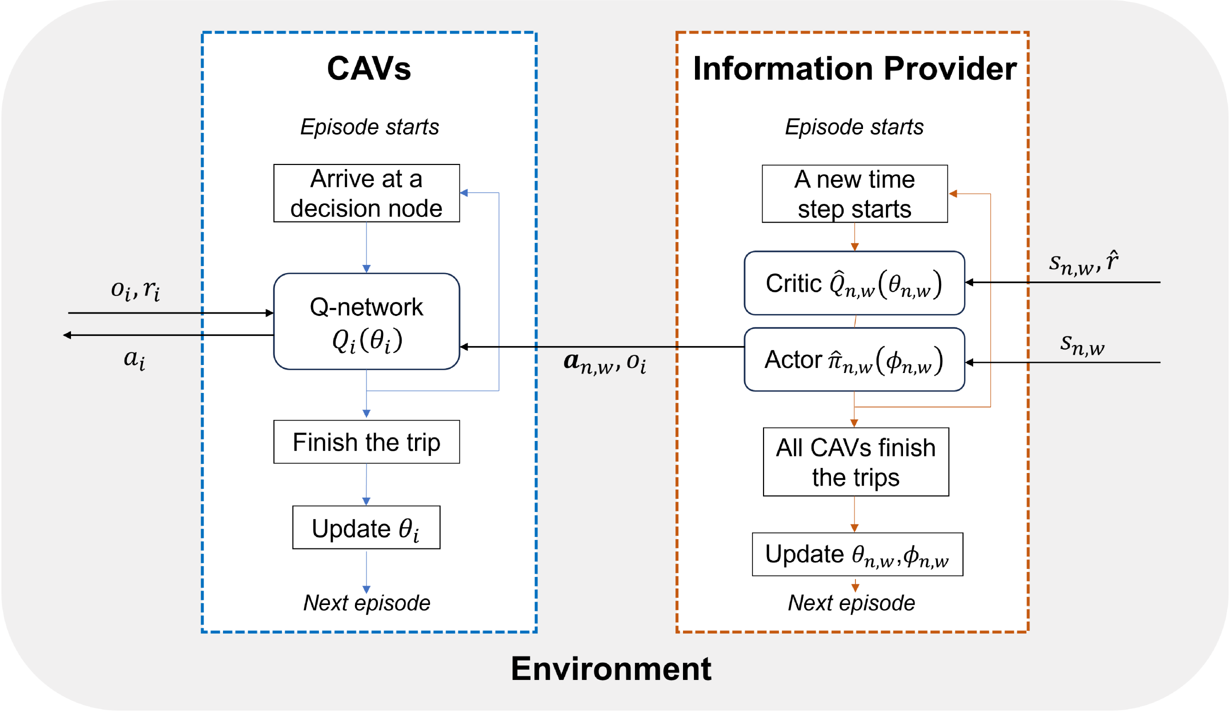

CAVs
Episode starts
Arrive at a decision node
Q-network $Q_i(\theta_i)$
Finish the trip
Update θ_i
Next episode
o_i, r_i
a_i
$\boldsymbol{a}_{n,w}, o_i$
Information Provider
Episode starts
A new time step starts
Critic $\hat{Q}_{n,w}(\theta_{n,w})$
Actor $\hat{\pi}_{n,w}(\phi_{n,w})$
All CAVs finish the trips
Update $\theta_{n,w}, \phi_{n,w}$
Next episode
$s_{n,w}, \hat{r}$
$s_{n,w}$
Environment

Figure 3: The two-way deep RL framework

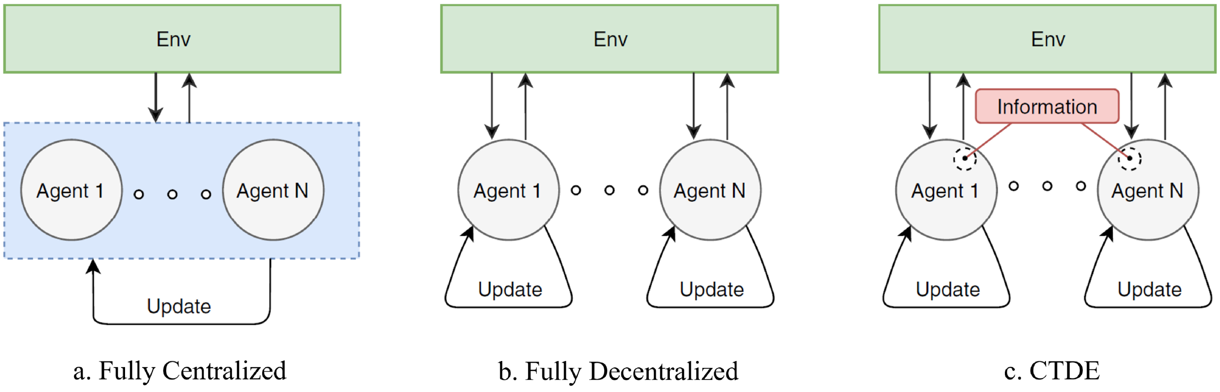
Since the problems formulated in this research have distinct characteristics, we customized a two-way RL framework in this study. In the following, we present the details of the multi-agent RL for CAVs and the single-agent RL for the information provider in Section 4.1 and Section 4.2, respectively. Then, a two-way deep RL algorithm is presented in Section 4.3.

4.1 Multi-agent RL for CAVs

In multi-agent RL, there usually presents two challenges: stability and the curse of dimensionality. First, since multiple agents learn concurrently, the environment faced by each individual agent is non-stationary. The reward and policy of one agent is affected by the joint behavior of other agents. Theoretically, if each individual agent ignores this issue and optimizes its own policy, assuming a stationary environment in a fully independent and decentralized way, the learning is easy to fail to converge (Zhang et al., 2021; Gronauer and Diepold, 2022). Second, as the number of agents and the size of state/action spaces increase, the learning becomes computationally intractable due to the curse of dimensionality (Zhang et al., 2021; Gronauer and Diepold, 2022).

In the literature on multi-agent RL, one solution to the above challenges is to use a Centralized Training and Decentralized Execution (CTDE) scheme, which allows information sharing among agents during the learning process. Figure 4 shows the three types of learning schemes in multi-agent RL. Figure 4(a) shows the fully centralized scheme where a central controller is required to maintain the joint policy and execute the joint action of all agents. Figure 4(b) shows the fully decentralized case where each agent interacts with the environment, executes actions, and updates policies independently. Figure 4(c) shows the CTDE scheme where agents can share some information, such as observations, rewards, or actions in the learning process, while they could execute actions independently.

The fully centralized scheme suffers from the curse of dimensionality as the central controller has to maintain the information of all agents and learn a joint policy. The fully decentralized scheme suffers from instability as it ignores the interaction and influences between agents. CTDE addresses the instability issue by allowing information exchange among agents so that they can take actions with the consideration of others' behaviors. Moreover, CTDE can also mitigate the dimensionality issue in training if some of the agents can share the learning experience or even share the Q-network, for example, in Q-learning. Also, the decentralized execution in CTDE does not require the agents to make joint actions and thus helps circumvent the dimensionality issue.



a. Fully Centralized b. Fully Decentralized c. CTDE

Figure 4: Learning schemes in multi-agent RL (Gronauer and Diepold, 2022)

This research follows a way similar to CTDE. However, we do not require strong assumptions on the information sharing, such as sharing actions, among all the agents. In fact, in the formulated Markov game for CAVs, we explicitly designed that the observation of each agent includes the real-time travel time information of each sub-route, which indirectly reflects the actions of some agents that may impact the agent. In this way, we do not require any strong assumptions on information sharing and maintain direct competition from the agents who act at the same time in the neighborhood. This helps mitigate the instability issue while maintaining realistic in practice.

Given that the observations of the CAV include both continuous values, i.e., the travel times of sub-routes, and discrete values, i.e., the time, location, and the suggested sub-route, we employ the deep Q-learning method. This method utilizes Deep Q-Networks (DQN) to approximate the Q functions. In particular, we leverage the Double Deep Q-Network (DDQN) to mitigate the overestimation of action values. Each agent maintains a Q-network $Q_i(o_i, a_i|\theta_i)$ parameterized by θ_i and a target Q-network $Q_i^-(\cdot|\theta_i^-)$ parameterized by θ_i^-. The parameter θ_i^- of the target Q-network is copied from the Q-network every several episodes to stabilize training. The DDQN updates the parameter θ_i by minimizing the following loss

$$\mathscr{L}(\theta_i) = \mathbb{E}_{o_i,a_i,o_i'}\left[r_i + \gamma Q_i^-\left(o_i',\ argmax_{a_i}\ Q_i(o_i', a_i|\theta_i)\mid\theta_i^-\right) - Q_i(o_i, a_i|\theta_i)\right]^2 \tag{6}$$

The agent learns the optimal $Q_i^*(o_i, a_i|\theta_i^*)$ gradually from its experience, and the optimal policy π_i^* is to choose the action that maximizes the optimal Q-value given an observation $o_i \in O_i$, i.e.,

$$\pi_i^*(o_i) = argmax_{a_i}\ Q_i^*(o_i, a_i|\theta_i^*) \tag{7}$$

As mentioned in Section 3.2, the Markov routing game of CAVs has characteristics differing from the standard Markov game; hence, customization has been made in the design of the Q-networks. Since the observation includes both continuous and discrete values, and some of them (the location and the suggested sub-route) are even categorical instead of numerical values, we must tailor the input layer of the Q-networks to enhance learning performance. Moreover, the varying size of action sets at different decision nodes adds additional challenges in designing the Q-networks.

To tackle these unique challenges, we designed separate but virtually connected sub-networks for each decision node given an OD pair. Figure 5 shows the network structure. Given an OD pair $w \in W$, the set of decision nodes D_w is fixed. The green nodes at the top of the figure represent different decision nodes. Each decision node has a sub-network, where the size of the action set is fixed. The bottom layer outputs the Q-values for each action. The sub-networks are virtually connected through decision nodes. These virtual connections are represented by dashed lines, indicating that there are no parameters to train but value transmission is allowed. The second row in the figure contains other inputs, including time, travel times of sub-routes, and the suggested sub-route. The network has two layers of inputs, with the first layer as a virtual layer. The specific structure of each sub-network is not necessarily the same but depends on the traffic network topology.

During the training process, for each batch sample with o and o', the sub-networks of decision nodes in o and o' are connected to allow value transmission so that the loss in Equation (6) can be calculated and minimized. While in each execution, only one sub-network, i.e., the one corresponding to the decision node where the agent takes an action, will be utilized. In this way, we can address the issue of varying size of action set in a trip, and at the same time, largely reduce the dimension of the input for a trip.

For example, if we have an OD pair with m decision nodes, associated with which we have $n_1, n_2, .., n_m$ sub-routes and $k_1, k_2, .., k_m$ links to choose from at each decision node respectively. If we insist on using a single network for a trip, then the dimension of the input is $[m$ (decision nodes, one-hot encoding) + 1

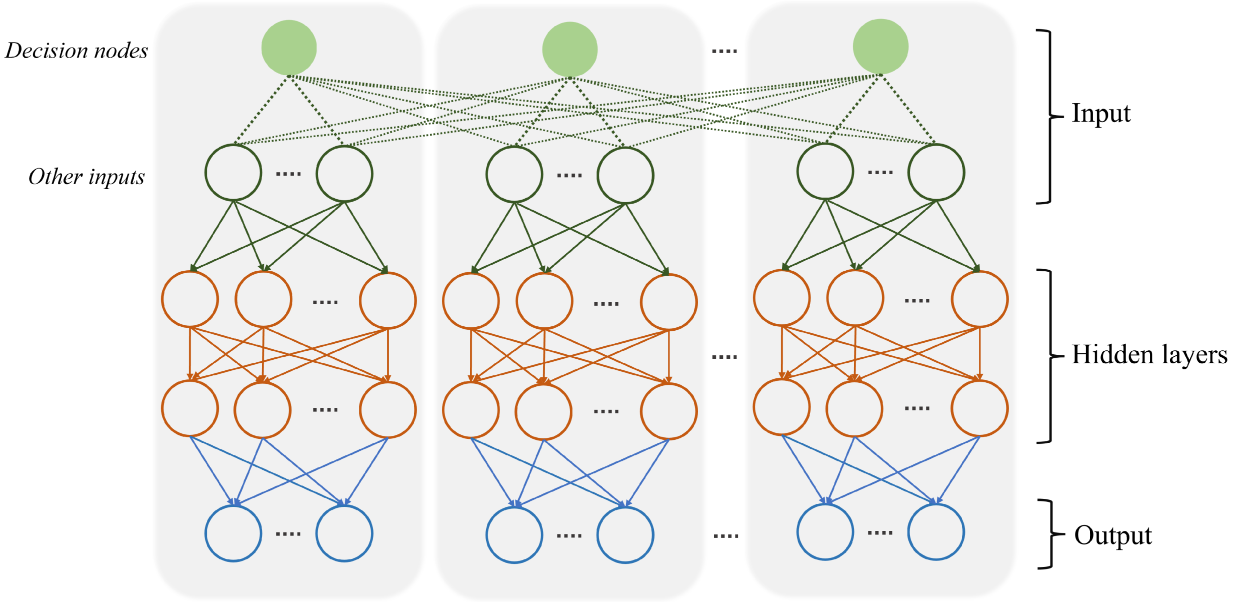


Figure 5: The Q-network structure

(time step) + $\sum_j n_j$ (travel times for sub-routes) + $\sum_j k_j$ (suggested sub-route, one-hot encoding)], where one-hot encoding is a common coding way for categorical values. In this case, the network contains too much redundant information that is irrelevant for a given decision node either for training or execution, not to mention the increase in the number of parameters to train. And we even need additional techniques to force the irrelevant actions for other decision nodes to be infeasible at the output layer for a single execution or fitting. However, using the virtually connected sub-networks, the input size reduces to $1+n_j+k_j$ for a single execution or fitting. Note that for the input dimension of suggested sub-routes, it is indeed the number of actions since multiple sub-routes can share the same link.

For better illustration, we use the Braess network shown in Figure 6 as an example. In the Braess network, we have one OD pair which is from node 1 to node 4, and $m=2$ decision nodes that are nodes 1 and 2. At node 1, we have $n_1=3$ sub-routes, which are routes 1-5, 2-3-4, and 2-4; and $k_1=2$ links to choose from, which are links 1 and 2. At node 2, we have $n_2=2$ sub-routes, which are sub-routes 3-5 and 4; and $k_2=2$ links to choose from, which are links 3 and 4. In this case, if we use the traditional single Q-network for a trip, we will have the input dimension as $2+1+3+2+2+2=12$, which contains too much redundant information that is irrelevant for a given decision node. If we use the virtually connected sub-networks, the input size for the network associated with decision node 1 reduces to $1+3+2=6$ and reduces to $1+2+2=5$ for decision node 2, which could save even a lot in an execution or fitting for more complex networks. Additionally, different sub-networks can handle different number of actions in the output layer, i.e., different number of links to choose at different decision nodes.

In practice, for the context of this research, the curse of dimensionality induced by a large number of agents can be addressed by advanced technologies such as edge computing to allow individual agents to do computation locally. Moreover, it is noteworthy that achieving convergence might face challenges for real-world large-scale networks, even when CAVs' actions and rewards are implicitly reflected in observations accessible to other CAVs. In such cases, our approach can be extended to a more aggregate level by allowing agents to share their information, including observations, rewards, and used actions. For example, agents could be divided into groups, permitting those within the same group to share a common experience buffer and Q-network, assuming that information can be readily transmitted among CAVs.

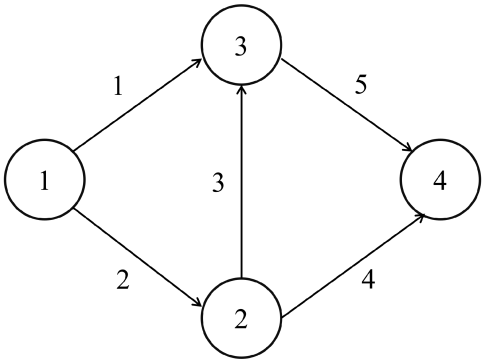


Figure 6: Braess network

4.2 Single-agent RL for the information provider

The information provider provides sub-route suggestions to CAVs given the real-time traffic state. In the MDP for the information provider, the action of the information provider is the probability distribution among the sub-routes available at a decision node $n \in D_w, w \in W$. According to this probability distribution, the information provider provides specific suggestions to different CAVs. Indeed, in computation, we convert it into the probability distribution among the links available at a decision node since multiple sub-routes can share the same link. The action of the information provider is a vector of continuous values, i.e., the probabilities of suggesting different links at a decision node. Therefore, we use the Deep Deterministic Policy Gradient (DDPG) method on the information provider side. Other methods for continuous action space can also be used.

DDPG is a RL approach that combines the power of deep neural networks (DNNs) with the stability of deterministic policy gradients. It is an actor-critic method for problems with continuous action spaces. DDPG uses DNNs to approximate both the actor (policy) and critic (value function). The actor network learns a deterministic policy by mapping states to continuous actions, while the critic network evaluates the quality of the chosen actions using a learned Q-value function.

For the critic network, we use the DQN method to approximate the action-values (Q-values), in which the loss function Equation (8) is minimized to update the parameters of the Q-network.

$$\mathscr{L}(\theta) = \mathbb{E}_{s,\boldsymbol{a},s'}\left[r + \gamma Q^{-}\left(s',\, argmax_{\boldsymbol{a}}\, Q(s',\boldsymbol{a}|\theta^{-}) \mid \theta^{-}\right) - Q(s,\boldsymbol{a}|\theta)\right]^2 \tag{8}$$

The difference is now the target Q-network is updated once per main Q-network update by Polyak averaging, instead of copying from the main Q-network every several episodes. Namely, do the following once per update:

$$\theta^{-} \leftarrow \rho\theta^{-} + (1-\rho)\theta$$

where ρ is a hyperparameter between 0 and 1.

The actor network π takes the current state as input and outputs a continuous action in the action space. The network is trained using gradient ascent to maximize the expected cumulative reward Equation (9), where ϕ is the parameter of the actor network. We also have a target actor network and update the parameters of the target actor network in the same averaging way as for the critic network.

$$max_{\phi}\mathbb{E}_s\left[Q(s, \pi(s \mid \phi) \mid \theta)\right] \tag{9}$$

The similar complexity also exists for the information provider side, as we have presented for the Markov game of CAVs, since the decision nodes vary among OD pairs and the action sets vary among decision nodes. To address this, we partition the problem into sub-problems by different ODs that share the same objective of minimizing the total system travel time. The same structure that separates the sub-networks for different decision nodes, as shown in Figure 5, is utilized for the actor and critic networks in each OD. However, we do not need to enable the value transmission among different sub-networks and can train the sub-networks independently. This also enables distributed learning on the information provider side in implementation. Additionally, to reduce the number of neural networks, OD pairs with the same destination can share the same actor and critic networks for any decision nodes they have in common, because the CAVs at a same decision node traveling to a same destination can be treated as homogeneous.

The RL on the information provider side also faces challenges caused by the non-fixed environment that involves CAVs who are learning at the same time. We utilize two techniques to help improve learning performance: reward shaping and periodic exploration.

Reward shaping is a technique used in RL to provide additional guidance to the learning agent. It involves designing a modified reward that encourages desired behaviors or simplifies the learning process. In this research, at the earlier stage of the learning process of the CAVs, they are unlikely to be influenced by the information provider as they explore the environment. And, at the same time, the information provider has not learned a desired policy and cannot estimate a fair approximation of Q-values as the CAVs do not behave in the suggested way. Hence, this may induce the information provider to provide suggestions that are unfavorable to the CAVs, which in turn jeopardizes the effect the information provider can have since CAVs will learn low Q-values for the action of following the suggestions. This reduces the learning performance of the information provider as the Q-values are highly uncertain and hard to learn.

To address this, we introduce an additional reward for the CAVs that follow the received suggestion at a decision node to encourage desired routing behaviors from the information provider's perspective. The total additional reward for following the suggestions at all encountered decision nodes, i.e., following a complete suggested route, is the same for all CAVs in all OD pairs. CAVs receive additional reward if they follow the suggestion at some but not all the decision nodes. Section 6.3 provides examples of the reward design. Moreover, Section 5 provides theoretical analysis regarding the additional reward. We will show that the additional reward has positive effects in reducing congestion. Note that, with the proposed framework, the resulting behaviors induced by the additional reward will not cause any sacrifice on individual travel times from CAVs' perspectives. This is in contrast to the traditional monetary incentives used for congestion mitigation, where travelers are compensated by the incentive for their sacrifice in travel time.

Periodic exploration is another technique that we use to improve learning performance. In RL, exploration is encouraged at the earlier stage of the learning process to help the agent explore the unknown environment. For example, ε-greedy is a common method used in DQN, which allows the agent to randomly take an action with a probability of ε. Such probability decays over time to help the learning converge. In DDPG, the common method used for exploration is to add noise to the action obtained from the current policy. The noise is also decayed over time by, for example, reducing the standard deviation of the noise.

In this research, since the policies of CAVs update over time, from the information provider's standpoint, the Q-values associated with previously explored state-action pairs may be outdated. It becomes necessary to revisit actions with low Q-values and update them with the latest estimated Q-values. This helps the information provider capture the policy updates of CAVs and maintains fair assessments of its own actions. To do this, we use periodic exploration which decays the noise within a certain period of time and then restores it to its original value in the new period and decays it again over time. When to restore the exploration is treated as a hyperparameter in this research.

4.3 Two-way deep reinforcement learning algorithm

This section presents the proposed two-way deep RL algorithm for CAVs and the information provider, as summarized in Algorithm 1. The proposed algorithm has incorporated all the unique characteristics of the models outlined in Section 3.

First, we initialize the Q-networks, critic, and actor networks, and the corresponding target networks. Also, initialize the experience replay buffers for CAVs and the information provider, which are used to store the collected experiences for batch training and updating the networks. Then there are two loops. One is the loop for episodes, and another is the loop for time steps within each episode. At the end, the algorithm outputs the learned networks, from which we obtain the learned policies.

In each episode, there are two main stages: (1) the CAVs and the information provider interact with the shared environment and collect experiences into buffers; (2) sample batches from the experience buffers to update the networks by minimizing the corresponding loss functions or maximizing the expected cumulative reward; along with certain adjustments to hyperparameters. In the loop of time steps, the information provider observes its state and takes actions, and stores experiences if it is not the initial time step. Each CAV only takes actions at a decision node, not necessarily at each time step. If the CAV arrives at the destination, it receives a reward and stores the experience, and then finishes the trip. The information provider provides en-route suggestions for the CAVs at a decision node and ends the episode once all CAVs finish their trips. On the information provider side, the learning is independent for each decision node but shares the same reward and objective. The transition of the environment is triggered at each time step, no matter how many CAVs took actions in the previous time step.

5 Correlated Equilibrium

The proposed two-way RL addresses the UO-SO conflict upon the theoretical foundation of CE. Section 5.1 introduces the CE using an illustrative example. Section 5.2 formulates the CE in the present research context and analyzes how the output of the proposed framework is consistent with the CE structure. Section 5.3 presents theoretical analysis of the proposed framework, including its ability to achieve CE, its effect on congestion mitigation, and the impact of reward shaping.

5.1 Introduction of CE

This section introduces the concept of CE and provides an illustrative example. The CE is defined as follows (Aumann, 1987).

Definition 3. *In a multi-player game, a trusted third party assigns an action to every player according to a certain joint probability distribution. If no player wants to deviate from the assigned action, as deviating from it will not receive better utility, assuming others do not deviate, then a correlated equilibrium is reached. The distribution from which the joint action is drawn is called the correlated equilibrium.*

Consider a game with a set of players $\mathcal{I} = \{1, 2, ..., I\}$. Let $\mathcal{S}$ denote the strategy set of the third party, with $\boldsymbol{s} \in \mathcal{S}$ being a single strategy profile. A strategy profile $\boldsymbol{s}$ stipulates the specific suggestions provided to different players. Let $\boldsymbol{s}_i \in \mathcal{S}_i$ be the action suggested to player i by the strategy profile $\boldsymbol{s}$, where $\mathcal{S}_i$ is the action set of player i. The third party suggests actions to every player according to a strategy profile

Algorithm 1: *Two-way deep reinforcement learning algorithm*

```
Input: Time step length $h$, maximum number of episode $E$ and maximum time horizon $t_{max}$
Initialization: Initialize networks $Q_i(\theta_i)$ and $Q_i^-(\theta_i^-)$ for each CAV $i$; and $\widehat{Q}_{n,w}(\theta_{n,w})$, $\widehat{\pi}_{n,w}(\phi_{n,w})$,
    $\widehat{Q}^-_{n,w}(\theta^-_{n,w})$, and $\widehat{\pi}^-_{n,w}(\phi^-_{n,w})$, $n \in D_w, w \in W$ for the Information Provider (IP)
Initialize the experience buffers $B_i$ for each CAV $i$, and $\widehat{B}_{n,w}, n \in D_w, w \in W$ for the IP
Set episode $e = 0$
while $e \leq E$ do
  Reset the environment. The IP gets an initial $s_{n,w,0}$; each CAV $i$ draws an initial $o_{i,0}$
  Set $done_i = 0$, latest execution time $\dot{t}_i = -1$ for each CAV $i$, and $t = 0$
  while $t \leq t_{max}$ do
    if $\exists\, i,\ done_i = 0$ then
      if $t > 0$ then
        The IP observes the state $s_{n,w,t}$, and receive a reward $\widehat{r}$
        The IP stores experience tuples $(s_{n,w,t-1}, \boldsymbol{a}_{n,w,t-1}, \widehat{r}, s_{n,w,t})$ into buffers $\widehat{B}_{n,w}, n \in D_w, w \in W$
      The IP executes action $\boldsymbol{a}_{n,w,t}$ according to $\widehat{\pi}_{n,w}$ and $s_{n,w,t}$, under noise
      for $i = 1$ to $I$ do
        if CAV $i$ arrives at its destination then
          $done_i = 1$
          CAV $i$ receives a reward $r_i$ and stores experience $(o_{i,\dot{t}_i}, a_{i,\dot{t}_i}, r_i, o_{i,t})$ into buffer $B_i$
          Continue
        else
          if CAV $i$ arrives at a decision node $n \in D_w, w \in W$ then
            The IP provides a sub-route suggestion to CAV $i$, according to $\boldsymbol{a}_{n,w,t}$
            CAV $i$ observes $o_{i,t}$
            if $\dot{t}_i \geq 0$ then
              CAV $i$ receives a reward $r_i$ and stores $(o_{i,\dot{t}_i}, a_{i,\dot{t}_i}, r_i, o_{i,t})$ into buffer $B_i$
            CAV $i$ executes $a_{i,t}$ using $\varepsilon$-greedy
            Set $\dot{t}_i = t$
    else
      The IP observes $s_{n,w,t}$, receives $\widehat{r}$, and stores $(s_{n,w,t-1}, \boldsymbol{a}_{n,w,t-1}, \widehat{r}, s_{n,w,t})$ into $\widehat{B}_{n,w}$
      Break
    Environment transition from $s_t$ to $s_{t+1}$, $t = t + 1$
  for $k_i = 1$ to $K_i$ do
    Sample a batch of experience tuples from the buffer $B_i, \forall i$
    Update the Q-network $Q_i(\theta_i), \forall i$
  for $\widehat{k} = 1$ to $\widehat{K}$ do
    Sample a batch of experience tuples from the buffer $\widehat{B}_{n,w}, \forall n \in D_w, w \in W$
    Update the critic network $\widehat{Q}_{n,w}(\theta_{n,w})$ and actor network $\widehat{\pi}_{n,w}(\phi_{n,w}), \forall n \in D_w, w \in W$
    Update the target networks $\widehat{Q}^-_{n,w}(\theta^-_{n,w})$ and $\widehat{\pi}^-_{n,w}(\phi^-_{n,w})$ by Polyak averaging
  if $e$ mod $m$ =0 then
    Copy the parameters of $Q_i(\theta_i)$ to the target network $Q_i^-(\theta_i^-), \forall i$
  Decrease the exploration rate $\varepsilon$ and learning rate of CAVs
  Decrease the learning rate of the IP and adjust the noise following the periodic exploration
  Set $e = e + 1$
Return Networks $Q_i(\theta_i), \forall i$, and $\widehat{Q}_{n,w}(\theta_{n,w})$, $\widehat{\pi}_{n,w}(\phi_{n,w}), n \in D_w, w \in W$
```

drawn from a probability distribution $\mathcal{P}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$. Based on the CE definition, the probability distribution $\mathcal{P}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$ is called a CE if no player can obtain higher utility by unilaterally deviating from the received suggestion. That is, the following holds:

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \left(u_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) - u_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i}) \right) \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{10}$$

where u_i is the utility of player i, $\boldsymbol{s}_i'$ is an action different from $\boldsymbol{s}_i$, and $\boldsymbol{s}_{-i}$ represents the actions of all other players except i.

CE is a generalization of Nash Equilibrium (NE), i.e., any NE must be a CE; CEs provide a richer set of solutions than NEs (Cigler and Faltings, 2011). The difference between CE and NE is that players behave independently to maximize their own payoffs in NE, while CE provides a mechanism with a third party involved for players to correlate their actions to arrive at mutually higher payoffs (Marris et al., 2021). The maximum total system payoff at CE is at least that any NE (Marris et al., 2021). And User Equilibrium (UE) must be a NE (Bell, 2000). Hence, CE offers the theoretical foundation to mitigate the UO-SO conflict in traffic systems since it provides the opportunity that the system and individual travelers both get better off compared to NE.

Below we use a small example on the Braess network for a better demonstration of CE. We do not consider en-route behaviors in this example, since CE is defined in a static form, and the purpose here is to illustrate the concept of CE. The information provider here is considered to be providing suggestions of a complete route. Figure 6 has shown the network structure previously. Assume we have 10 travelers departing at the same time from node 1 to node 4. The link travel times are determined by predefined fundamental diagrams. We use triangular fundamental diagrams in this example. The link characteristics are shown in Table 1. We do not specify the units of the link characteristics and use small numbers to be paired with the small travel demand of 10 travelers for the ease of manual calculation.

Table 1: Link characteristics of Braess network.

Links	Free flow travel time	Capacity	Critical density	Jam density	Length
1	5	1	5	20	1
2	1	3	3	12	1
3	1	3	3	12	1
4	4	1	4	16	1
5	3	2	6	24	1

Consider route 1 contains links 1 and 5, route 2 contains links 2, 3, and 5, and route 3 contains links 2 and 4. For this example, we can calculate the route flow at UE is $[3,3,4]$, representing the route flows on routes 1, 2, and 3, respectively. And the travel time at UE is calculated as 8 for all the routes.

Now, consider we have a third party who is an information provider providing route suggestions. The information provider assigns route suggestions to travelers according to a joint probability distribution $\mathcal{P}(\boldsymbol{s})$. An example of the joint probability distribution is shown in Table 2. Each row represents a single strategy profile that is a combination of suggested routes to the 10 travelers. The table indicates to provide the first joint suggestions with probability $\frac{1}{4}$, the second with $\frac{1}{4}$, and the third with $\frac{1}{2}$.

This is a CE since no traveler can be better off by unilaterally deviating the received suggestion. Take traveler 1 as an example. If traveler 1 receives a suggestion of route 1, then we know either the first combination or the third combination of route choices will be realized. The probability of the first combination

Table 2: An example of route suggestion distribution.

Probability	Travelers									
	1	2	3	4	5	6	7	8	9	10
1/4	1	1	1	2	2	2	3	3	3	3
1/4	2	3	3	1	1	3	3	1	2	2
1/2	1	1	1	1	1	2	3	3	3	3

is $\frac{1}{3}$ and that of the third one is $\frac{2}{3}$, on the condition that the second combination must not happen. Then, if the traveler 1 deviates to route 2, to finish the trip, she(he) will need 11 units of travel time when the first combination happens and 7 when the third combination happens. If traveler 1 deviates to route 3, she(he) will need 11 units of travel time when the first combination happens and 8 when the third combination happens. Hence, the expected travel time of deviating to route 2 is $11\times\frac{1}{3}+7\times\frac{2}{3}=8.33\geq 8$. And the expected travel time for deviating to route 3 is $11\times\frac{1}{3}+8\times\frac{2}{3}=9\geq 8$. Therefore, traveler 1 cannot get better off when deviating from the suggested route. Similarly, we can calculate the case where traveler 1 receives a suggestion of route 2. The expected travel time for deviating to route 1 is 8 and 9 for deviating to route 3. Hence, traveler 1 cannot get better off when deviating from the suggested route, either. One can verify this for all other travelers by simple calculations.

At this CE, the expected total travel time is $\frac{1}{4}\times 80+\frac{1}{4}\times 80+\frac{1}{2}\times 70=75$, which is better than UE where the total travel time is $8\times 10=80$.

At the individual level, take traveler 1 and traveler 10 as examples. The expected travel time is $\frac{1}{4}\times 8+\frac{1}{4}\times 8+\frac{1}{2}\times 8=8$ for traveler 1 and $\frac{1}{4}\times 8+\frac{1}{4}\times 8+\frac{1}{2}\times 6=7$ for traveler 10. One can easily calculate the values for other travelers and find that all the travelers either get better off or get equal travel time compared to UE. A natural question may arise about the fairness issue since different travelers may receive different expected travel time. This inequality can be avoided by altering the order of travelers in Table 2 in a rotated way, allowing equal probabilities for all travelers to access any route. In fact, following such a manner generates another CE that has the same expected total travel time with the one in Table 2 but has equal expected individual travel time for all travelers. Each CE, or a joint probability distribution in general, with unequal expected individual travel time has at least a counterpart that ensures such fairness by the previously mentioned way. However, doing so will largely increase the number of strategy profiles in $\mathcal{S}$, which poses challenges for large-scale problems. Section 5.2 will show that, with the proposed framework, equal expected individual travel time can always be ensured for travelers departing at the same time for the same OD, addressing the challenge raised by the increased joint strategy set in order to ensure fairness.

5.2 CE in the proposed framework

This section formulates the CE within the proposed research context, presents how the probability distribution of en-route suggestions learned by the information provider is consistent with the structure of CE, and discusses its features.

First, we formulate the CE in the context of this research. While the information provider provides en-route suggestions at decision nodes, such suggestions at the end can be realized as one suggestion of a complete route for each CAV. Consistent with the original definition, the CE in this research is represented by a joint probability distribution $\mathcal{P}(\boldsymbol{s}),\boldsymbol{s}\in\mathcal{S}$. Denote the set of CAVs as $\mathcal{I}=\{1,2,...,I\}$. The strategy set of the information provider is $\mathcal{S}$. Each single strategy profile $\boldsymbol{s}\in\mathcal{S}$ is a set of joint route suggestions provided to CAVs. $\boldsymbol{s}_i \in \mathcal{S}_i$ is the route suggested to CAV i by $\boldsymbol{s}$, with $\mathcal{S}_i$ being the set of available routes for CAV i. The information provider's joint route suggestion $\boldsymbol{s}$ follows a probability distribution $\mathcal{P}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$. A distribution $\mathcal{P}(\boldsymbol{s})$ is a CE if Equation (10) holds. For reader's convenience, we rewrite it in the following.

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \left(u_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) - u_i(\boldsymbol{s}'_i, \boldsymbol{s}_{-i}) \right) \geq 0, \forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{11}$$

where u_i is the utility of CAV i, $\boldsymbol{s}'_i$ is a route different from $\boldsymbol{s}_i$, $\boldsymbol{s}_{-i}$ represents the route choices of all other players except i. Here, because we consider reward shaping that involves an additional reward for CAVs following the received suggestions, the utility is formulated as follows.

$$u_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) = -\tau_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) + \frac{\eta}{\alpha}, \qquad u_i(\boldsymbol{s}'_i, \boldsymbol{s}_{-i}) = -\tau_i(\boldsymbol{s}'_i, \boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \times \beta_{\boldsymbol{s}_i, \boldsymbol{s}'_i} \tag{12}$$

where τ_i represents the travel time of CAV i, η is the additional reward for following the en-route suggestions at all decision nodes (following a complete route), α is the value of travel time, and $0 \leq \beta_{\boldsymbol{s}_i, \boldsymbol{s}'_i} < 1$ is the proportion of the additional reward η that the CAV can get for using $\boldsymbol{s}'_i$ with the suggestion being $\boldsymbol{s}_i$. $\eta = 0$ is a special case when no additional reward is provided.

The concept of CE is defined in a static context where the provided suggestion is a complete route, while suggestions are provided in an en-route manner in our proposed framework. Hence, in the following, we demonstrate that the action of the information provider, i.e., the probabilities of suggesting different links at decision nodes at different time steps, can derive a unique joint probability distribution of complete-route suggestions.

Consider a general case where the adopted RL algorithm of the information provider learns stochastic policies, i.e., the actor neural network outputs the parameters of the distribution of action and the actual action is obtained by sampling based on the learned parameters of the distribution. If an RL algorithm that learns deterministic policies is adopted, it is a special case of the general stochastic one and the following analysis is also applicable.

First, the action $\boldsymbol{A} = \{\widehat{\boldsymbol{a}}_{n,w}, n \in D_w, w \in W\}$ made at all different time steps $t \in T$ can be realized as a static profile containing the probabilities of suggesting different complete routes, denoted by $\boldsymbol{p}_{w,t}, w \in W, t \in T$, where the dimension of $\boldsymbol{p}_{w,t}$ is the number of routes (J_w) connecting OD pair $w \in W$. Then $\boldsymbol{p}_{w,t}(j) = \prod_{\{a,n|a \in j, a \in A_{n,w}, n \in D_w\}} \widehat{\boldsymbol{a}}_{n,w}(a)$, for route j in $w \in W$, representing the probability of being suggested route j for CAVs departing at $t \in T$ and traveling in $w \in W$. Here in $\boldsymbol{p}_{w,t}$, we do not need to differentiate $\widehat{\boldsymbol{a}}_{n,w}$ by time step because the CAVs departing at the same time that use a same route j must arrive at any decision node simultaneously and encounter the same $\widehat{\boldsymbol{a}}_{n,w}$. Therefore, just use $\widehat{\boldsymbol{a}}_{n,w}(a)$ to represent the probability of being suggested a at the time when the CAVs arrive at decision node n. Denote the joint probability distribution that assigns the routes to CAVs based on $\boldsymbol{p}_{w,t}$ as $\mathcal{P}^{\boldsymbol{A}}_{w,t}(\boldsymbol{s}^{w,t}), \boldsymbol{s}^{w,t} \in \mathcal{S}^{\boldsymbol{A}}_{w,t}, w \in W, t \in T$, where $\boldsymbol{s}^{w,t}$ is a joint route suggestion profile that assigns routes to CAVs departing at t in OD pair w. Here, the total number of the joint route suggestion profiles with positive probability in $\mathcal{S}^{\boldsymbol{A}}_{w,t}$ is

$$\prod_{j=1}^{J_w} \binom{d_{w,t} - \sum_{i=1}^{j-1} \boldsymbol{p}_{w,t}(i) \times d_{w,t}}{\boldsymbol{p}_{w,t}(j) \times d_{w,t}},$$

representing all possible route assignments to different CAVs, where $d_{w,t}$ is the number of CAVs departing at t in OD pair w. And $\mathcal{P}^{\boldsymbol{A}}_{w,t}(\boldsymbol{s}^{w,t})$ assigns equal probabilities to each $\boldsymbol{s}^{w,t} \in \mathcal{S}^{\boldsymbol{A}}_{w,t}$. Denote the probability distribution of $\boldsymbol{A}$ output by the stochastic policy as $\mathcal{P}'(\boldsymbol{A}), \boldsymbol{A} \in \mathcal{A}$. For continuous RL algorithms, $\mathcal{P}'(\boldsymbol{A})$ can be estimated by taking the integral of the corresponding probability density function over a small interval around $\boldsymbol{A}$ that can be determined by a pre-designed precision. Now, the joint probability distribution $\mathcal{P}(\boldsymbol{s}) = \mathcal{P}'(\boldsymbol{A})\prod_{w\in W,t\in T}\mathcal{P}^{\boldsymbol{A}}_{w,t}(\boldsymbol{s}^{w,t}), \boldsymbol{s}\in\mathcal{S}$, where $\mathcal{S}=\bigcup_{\boldsymbol{A}\in\mathcal{A}}\prod_{w\in W,t\in T}\mathcal{S}^{\boldsymbol{A}}_{w,t}$, is the unique equivalent of the action $\boldsymbol{A}$ at all $t\in T$ output by the learned policy of the information provider.

So far, we have demonstrated how the action of the information provider at all decision nodes and all time steps can be converted into a unique equivalent joint probability distribution of complete-route suggestions. Note that, with the proposed framework, the $\mathcal{P}(\boldsymbol{s}), \boldsymbol{s}\in\mathcal{S}$ learned by the information provider always ensures all the CAVs departing at the same time for the same OD have the same expected travel time. This is because the nature of the information provider's action, which is the probabilities of suggesting different en-route choices, inherently makes the route assignment completely random with no differentiation among CAVs at a decision node. This feature results in the equal probability assigned to each $\boldsymbol{s}^{w,t}\in\mathcal{S}^{\boldsymbol{A}}_{w,t}$ in $\mathcal{P}^{\boldsymbol{A}}_{w,t}(\boldsymbol{s}^{w,t})$. As mentioned before, ensuring such fairness in a joint probability distribution can greatly increase the number of joint route suggestion profiles compared to its counterpart with the same expected total travel time. This poses challenges for large-scale problems. In fact, the design of the action of the information provider in the proposed framework exactly addresses this issue, by learning the probability distribution of en-route suggestions at different decision nodes instead of dealing with a complicated joint probability distribution. It ensures the fairness, circumvents the dimensionality challenge of CE, and meanwhile captures the en-route and dynamical nature of the routing behavior.

In the following analysis in Section 5.3, to avoid repeated description, we directly use the equivalent joint probability distribution of route suggestions as the representation of the information provider's output actions for convenience.

5.3 Theoretical analysis

This section conducts a theoretical analysis of the CE in the proposed framework, the effect on congestion mitigation, and the impact of the reward shaping.

The analysis of this section is based on the premise that the adopted RL algorithms are effective in the two-way learning framework. We stipulate that an RL algorithm is effective if it converges to the optimal policy. Within the proposed framework, different RL algorithms can be adopted and the following analysis is independent of them.

Theorem 1. *The proposed two-way RL framework must converge to a correlated equilibrium, given effective RL algorithms.*

Proof. Let $\mathcal{P}^*(\boldsymbol{s}), \boldsymbol{s}\in\mathcal{S}$ be the joint probability distribution learned by the information provider at the converged solution. With effective RL algorithms, the policy learned by each CAV i, π^*_i, is the best response to π^*_{-i}, where π^*_{-i} is the learned policies of all other CAVs except i. That is, the the state-values satisfy $V^*_{i,(\pi^*_i,\pi^*_{-i})}(s)\geq V^*_{i,(\pi_i,\pi^*_{-i})}(s), \forall\pi_i\neq\pi^*_i$, for $s\in S, i\in\mathcal{I}$. The expected cumulative reward of CAV i with the optimal policy π^*_i is $V^*_{i,(\pi^*_i,\pi^*_{-i})}(s_0)=\mathbb{E}_{\boldsymbol{s}_{-i}\sim\mathcal{P}^*(\cdot|\boldsymbol{s}_i)}(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i})+\eta/\alpha)$. Because the expected cumulative reward is optimized by each CAV, i.e., $V^*_{i,(\pi^*_i,\pi^*_{-i})}(s_0)\geq V^*_{i,(\pi_i,\pi^*_{-i})}(s_0), \forall\pi_i\neq\pi^*_i$, the following inequality holds $\forall\boldsymbol{s}'_i\neq\boldsymbol{s}_i, \forall\boldsymbol{s}_i\in\mathcal{S}_i, \forall i\in\mathcal{I}$.

$$\mathbb{E}_{\boldsymbol{s}_{-i}\sim\mathcal{P}^*(\cdot|\boldsymbol{s}_i)}(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i})+\eta/\alpha)\geq\mathbb{E}_{(\boldsymbol{s}_i,\boldsymbol{s}_{-i})\sim\mathcal{P}^*(\boldsymbol{s})}(-\tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i})+\frac{\eta}{\alpha}\times\beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i}) \tag{13}$$

That is,

$$\mathbb{E}_{\boldsymbol{s}_{-i}\sim\mathcal{P}^*(\cdot|\boldsymbol{s}_i)}(u_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i})-u_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}))\geq 0, \forall\boldsymbol{s}'_i\neq\boldsymbol{s}_i, \forall\boldsymbol{s}_i\in\mathcal{S}_i, \forall i\in\mathcal{I} \tag{14}$$

which can be written as

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})(u_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) - u_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i})) \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{15}$$

This is exactly consistent with the definition of CE. ■

Denote the sets of NE and CE as Ω_{NE} and Ω_{CE}, respectively, where only travel time is considered in the utility function. Further denote the set of CE, where both travel time and additional reward for following suggestions are considered in the utility function, as $\Omega_{\overline{CE}}$. We have the following theorem.

Theorem 2. $\Omega_{NE} \subseteq \Omega_{CE} \subseteq \Omega_{\overline{CE}}$ *holds. And the total travel time at the converged equilibrium by the proposed framework is no larger than the total travel time at the Nash equilibrium (or the minimum total travel time of Nash equilibria, if multiple NE exist).*

Proof. Based on the definition of CE, the following inequality holds for any joint probability distribution $\mathcal{P}(\boldsymbol{s}) \in \Omega_{CE}$:

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i})) \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{16}$$

Introduce an additional reward to Equation (16) for CAVs following the suggestions. The following inequality must hold since $0 \leq \beta_{\boldsymbol{s}_i,\boldsymbol{s}_{-i}} < 1$.

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}_i'} \right) \right) \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{17}$$

which implies $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ based on Equation (11) and Equation (12). Therefore, for any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{CE}$, we have $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$.

Based on Equation (17), the following holds for any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$.

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i}) + (1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}_i'})\frac{\eta}{\alpha} \right) = \delta_{\boldsymbol{s}_i,\boldsymbol{s}_i'} \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{18}$$

Namely, for $\forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$, we have

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i}) \right) + \sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}_i'})\frac{\eta}{\alpha} = \delta_{\boldsymbol{s}_i,\boldsymbol{s}_i'} \geq 0 \tag{19}$$

Based on Equation (19), if $\delta_{\boldsymbol{s}_i,\boldsymbol{s}_i'} - \sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}_i'})\eta/\alpha \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$, then we have $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i}) \right) \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$, which implies $\mathcal{P}(\boldsymbol{s}) \in \Omega_{CE}$. Otherwise, there exist $\boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$ such that $\delta_{\boldsymbol{s}_i,\boldsymbol{s}_i'} - \sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}_i'})\eta/\alpha < 0$, i.e., $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i',\boldsymbol{s}_{-i}) \right) < 0$, which implies $\mathcal{P}(\boldsymbol{s}) \notin \Omega_{CE}$. Therefore, there can exist $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ while $\mathcal{P}(\boldsymbol{s}) \notin \Omega_{CE}$. In particular, if $\eta = 0$, any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ is also in Ω_{CE}, and in fact, $\Omega_{CE} = \Omega_{\overline{CE}}$ when $\eta = 0$.

So far, we have proved $\forall \mathcal{P}(\boldsymbol{s}) \in \Omega_{CE}, \mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ holds, and there can exist $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ with $\mathcal{P}(\boldsymbol{s}) \notin \Omega_{CE}$. Therefore, $\Omega_{CE} \subseteq \Omega_{\overline{CE}}$ holds, and $\Omega_{CE} = \Omega_{\overline{CE}}$ when $\eta = 0$.

We have known $\Omega_{NE} \subseteq \Omega_{CE}$. Hence, $\Omega_{NE} \subseteq \Omega_{CE} \subseteq \Omega_{\overline{CE}}$ holds. Therefore, the minimum total travel time in $\Omega_{\overline{CE}}$ is not larger than the total travel time at any NE. Effective RL algorithms ensure the learning to converge to the optimal CE that minimizes the total travel time in $\Omega_{\overline{CE}}$. ■

Next, we analyze the impact of the additional reward η.

Theorem 3. *The minimum total travel time the two-way learning can achieve, i.e., the lower bound of the total travel time of a CE, either decreases or remains unchanged as the additional reward η increases.*

Proof. This theorem holds true as long as $\Omega_{\overline{CE}-\eta} \subseteq \Omega_{\overline{CE}-\eta'}$ for any $\eta' > \eta$. For any $\mathcal{P}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$,

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} + \frac{\eta'-\eta}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}) + \left(\frac{\eta}{\alpha} + \frac{\eta'-\eta}{\alpha} \right) \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) \tag{20}$$

can be written as

$$\begin{aligned} &\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) \\ &+ \sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\frac{\eta'-\eta}{\alpha} - \left(\frac{\eta'-\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) \end{aligned} \tag{21}$$

Because $0 \leq \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} < 1$, $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\frac{\eta'-\eta}{\alpha} - \left(\frac{\eta'-\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) > 0$. And for any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}-\eta}$, Equation (17) holds. Hence, Equations (21) and (20) must be larger than zero. Namely,

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \frac{\eta'}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}) + \frac{\eta'}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) > 0, \forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{22}$$

which implies, $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}-\eta'}$ must hold for any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}-\eta}$.

Now, for any $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}-\eta'}$, let the value of the left-hand side of Equation (22), i.e., the value of Equation (21), be denoted as $\delta_{\boldsymbol{s}_i,\boldsymbol{s}'_i}$. Then when there exists any $\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}$ with $\delta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} - \sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\frac{\eta'-\eta}{\alpha} - \left(\frac{\eta'-\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) < 0$, we have

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \times \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \right) \right) < 0 \tag{23}$$

so that $\mathcal{P}(\boldsymbol{s}) \notin \Omega_{\overline{CE}-\eta}$. Namely, there can exist $\mathcal{P}(\boldsymbol{s}) \in \Omega_{\overline{CE}-\eta'}$ that makes $\mathcal{P}(\boldsymbol{s}) \notin \Omega_{\overline{CE}-\eta}$ hold. Hence, we have $\Omega_{\overline{CE}-\eta} \subseteq \Omega_{\overline{CE}-\eta'}$. ■

The next theorem explores how the system optimum can be achieved and the relationship between the additional reward η and the potential in achieving SO. Here the system optimum is represented as a joint probability distribution that has the minimized expected total travel time. In general, the joint probability distribution at SO can be non-unique, but the one achieved by the proposed framework is unique because, as mentioned in Section 5.2, the nature of the information provider's action always results in a unique equivalent joint probability distribution with fairness ensured and no loss in optimality.

Theorem 4. *The system optimum can be achieved if the additional reward η is not less than a certain bound $\underline{\eta}^{SO} = \max_{\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha \delta^{SO}_{\boldsymbol{s}_i,\boldsymbol{s}'_i} / (1 - \beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i})$. The relationships among the optimal CE in $\Omega_{\overline{CE}}$, denoted as $\mathcal{P}^*(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$, the system optimum $\mathcal{P}^{SO}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$, and the lower bound of additional reward ($\underline{\eta}$) that is needed for the two-way learning to converge to $\mathcal{P}^*(\boldsymbol{s})$ or $\mathcal{P}^{SO}(\boldsymbol{s})$, can be summarized into the following mutually exclusive cases.*

(A). If $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$, then $\mathcal{P}^*(\boldsymbol{s}) = \mathcal{P}^{SO}(\boldsymbol{s})$ and $\underline{\eta} = \underline{\eta}^{SO} = 0$ for converging to $\mathcal{P}^{SO}(\boldsymbol{s})$.

(B). If $0 \leq \eta < \underline{\eta}^{SO}$, then $\mathcal{P}^{SO}(\boldsymbol{s}) \notin \Omega_{\overline{CE}}$ and $\mathcal{P}^*(\boldsymbol{s}) \neq \mathcal{P}^{SO}(\boldsymbol{s})$. In this case, we have the following: (a) if $\mathcal{P}^*(\boldsymbol{s}) \in \Omega_{CE}$, then $\underline{\eta} = 0$; (b) otherwise, $\mathcal{P}^*(\boldsymbol{s}) \in (\Omega_{\overline{CE}} - \Omega_{CE})$. $\eta \geq \underline{\eta} = \max_{\boldsymbol{s}_i' \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha \delta^*_{\boldsymbol{s}_i, \boldsymbol{s}_i'}/(1-\beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'}) > 0$, for converging to $\mathcal{P}^*(\boldsymbol{s})$.

(C). If $\eta \geq \underline{\eta}^{SO} > 0$, then $\mathcal{P}^{SO}(\boldsymbol{s}) \in (\Omega_{\overline{CE}} - \Omega_{CE})$, $\mathcal{P}^*(\boldsymbol{s}) = \mathcal{P}^{SO}(\boldsymbol{s})$. $\underline{\eta} = \underline{\eta}^{SO} > 0$, for converging to $\mathcal{P}^{SO}(\boldsymbol{s})$.

Proof. For a given problem with $\mathcal{P}^{SO}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}$, the following Equation (24), representing $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$, must be valid for the learning to converge to $\mathcal{P}^{SO}(\boldsymbol{s})$.

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \left(\left(-\tau_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \right) - \left(-\tau_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i}) + \frac{\eta}{\alpha} \times \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'} \right) \right) \geq 0 \quad \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I} \tag{24}$$

That is,

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i}) \right) + \sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i})(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'}) \frac{\eta}{\alpha} \geq 0 \tag{25}$$

Let $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \left(-\tau_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i}) \right) = \delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}$. Equation (25) holds if η satisfies

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i})(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'}) \frac{\eta}{\alpha} \geq -\delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{26}$$

That is,

$$(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'}) \frac{\eta}{\alpha} \sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) \geq -\delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{27}$$

Because given any $\boldsymbol{s}_i \in \mathcal{S}_i$, $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) = 1$, then Equation (27) becomes

$$\eta \geq -\frac{\alpha \delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}}{(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'})}, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{28}$$

It implies that Equation (28) holds as long as the following Equation (29) holds.

$$\eta \geq \max_{\boldsymbol{s}_i' \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\frac{\alpha \delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}}{(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'})} \tag{29}$$

Therefore, the lower bound of the additional reward is $\underline{\eta}^{SO} = \max_{\boldsymbol{s}_i' \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha \delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}/(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'})$ for converging to $\mathcal{P}^{SO}(\boldsymbol{s})$.

Now we prove the three cases (A), (B), and (C).

For (A), it is straightforward that $\mathcal{P}^*(\boldsymbol{s}) = \mathcal{P}^{SO}(\boldsymbol{s})$ given $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$. Since $\Omega_{NE} \in \Omega_{CE}$, we have $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{CE}$. This implies $\delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'} \geq 0, \forall \boldsymbol{s}_i' \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$. Hence, $\max_{\boldsymbol{s}_i' \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha \delta^{SO}_{\boldsymbol{s}_i, \boldsymbol{s}_i'}/(1 - \beta_{\boldsymbol{s}_i, \boldsymbol{s}_i'}) = 0$, i.e., $\underline{\eta} = \underline{\eta}^{SO} = 0$.

For (B), because $\eta < \underline{\eta}^{SO}$, Equation (29) does not hold. Hence, $\mathcal{P}^{SO}(\boldsymbol{s}) \notin \Omega_{\overline{CE}}$ and $\mathcal{P}^*(\boldsymbol{s}) \neq \mathcal{P}^{SO}(\boldsymbol{s})$. To find the lower bound $\underline{\eta}$ for converging to $\mathcal{P}^*(\boldsymbol{s})$, following a similar logic in deriving

$\underline{\eta}^{SO}$, replace $\mathcal{P}^{SO}(\boldsymbol{s})$ and $\delta^{SO}_{\boldsymbol{s}_i,\boldsymbol{s}'_i}$ with $\mathcal{P}^{*}(\boldsymbol{s})$ and $\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i}$, respectively, in Eqs. (24)-(29). We can obtain $\underline{\eta} = \max_{\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i}/(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i})$ for converging to $\mathcal{P}^{*}(\boldsymbol{s})$. Within (B), if (a) $\mathcal{P}^{*}(\boldsymbol{s}) \in \Omega_{CE}$, we have $\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \geq 0, \forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$. Hence, $\max_{\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i}/(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i}) = 0$, i.e., $\underline{\eta} = 0$. If (b) $\mathcal{P}^{*}(\boldsymbol{s}) \in (\Omega_{\overline{CE}} - \Omega_{CE})$, then $\mathcal{P}^{*}(\boldsymbol{s}) \notin \Omega_{CE}$. There must exist $\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}$ that satisfies $\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i} < 0$. It implies $\max_{\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}} -\alpha\delta^{*}_{\boldsymbol{s}_i,\boldsymbol{s}'_i}/(1-\beta_{\boldsymbol{s}_i,\boldsymbol{s}'_i}) > 0$.

For (C), because $\eta \geq \underline{\eta}^{SO}$, $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{\overline{CE}}$ holds. Since $\underline{\eta}^{SO} > 0$, there must exist $\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \boldsymbol{s}_i \in \mathcal{S}_i, i \in \mathcal{I}$ that satisfies $\delta^{SO}_{\boldsymbol{s}_i,\boldsymbol{s}'_i} < 0$. This implies $\mathcal{P}^{SO}(\boldsymbol{s}) \notin \Omega_{CE}$. Therefore, we have $\mathcal{P}^{SO}(\boldsymbol{s}) \in (\Omega_{\overline{CE}} - \Omega_{CE})$ and $\mathcal{P}^{*}(\boldsymbol{s}) = \mathcal{P}^{SO}(\boldsymbol{s})$. $\underline{\eta} = \underline{\eta}^{SO} > 0$, for converging to $\mathcal{P}^{SO}(\boldsymbol{s})$.

Last, we prove that the union of the three cases, (A), (B), and (C), covers the complete feasible universe, with each case being mutually exclusive. To begin with, Case (A) covers all possibilities of the situation $\underline{\eta}^{SO} = 0$, because $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$ is equivalent to $\underline{\eta}^{SO} = 0$. To prove this, we only need to prove that if $\underline{\eta}^{SO} = 0$, $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$ must hold, since we have proved that $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$ is a sufficient condition of $\underline{\eta}^{SO} = 0$ when proving Case (A). Now, any $\boldsymbol{s} \in \mathcal{P}^{SO}(\boldsymbol{s})$ must be the system-optimal traffic assignment at the aggregate level; otherwise, the expected total travel time of $\mathcal{P}^{SO}(\boldsymbol{s})$ is not minimized, which contradicts with the definition of $\mathcal{P}^{SO}(\boldsymbol{s})$. Then, no matter how $\boldsymbol{s} \in \mathcal{S}$ is distributed by $\mathcal{P}^{SO}(\boldsymbol{s})$, $(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}))$ is the same for any $\boldsymbol{s}_{-i}$ given a $\boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$. That is,

$$\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i})\right) = -\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}), \forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{30}$$

Because $\underline{\eta}^{SO} = 0$, $\delta^{SO}_{\boldsymbol{s}_i,\boldsymbol{s}'_i} \geq 0$, i.e., $\sum_{\boldsymbol{s}_{-i}} \mathcal{P}^{SO}(\boldsymbol{s}_i,\boldsymbol{s}_{-i})\left(-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i})\right) \geq 0$, $\forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$. Hence,

$$-\tau_i(\boldsymbol{s}_i,\boldsymbol{s}_{-i}) + \tau_i(\boldsymbol{s}'_i,\boldsymbol{s}_{-i}) \geq 0, \forall \boldsymbol{s}'_i \neq \boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{S}_i, \forall i \in \mathcal{I} \tag{31}$$

which implies $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$ since Equation (31) is exactly consistent with the definition of NE. Therefore, $\mathcal{P}^{SO}(\boldsymbol{s}) \in \Omega_{NE}$ is equivalent to $\underline{\eta}^{SO} = 0$, and Case (A) covers all possibilities of the situation of $\underline{\eta}^{SO} = 0$.

The situation of $\underline{\eta}^{SO} > 0$ is divided into (B) and (C) based on the relationship between η and $\underline{\eta}^{SO}$. Therefore, the union of the three cases, (A), (B), and (C), covers all feasible possibilities, and (A) is exclusive to either (B) or (C). With $\underline{\eta}^{SO} > 0$, Case (B) and Case (C) are for the conditions $0 \leq \eta < \underline{\eta}^{SO}$ and $\eta \geq \underline{\eta}^{SO}$, respectively. This means Case (B) and Case (C) cover the entire situation of $\underline{\eta}^{SO} > 0$ and at the same time are exclusive to each other. Hence, the three cases, (A), (B), and (C), cover the complete feasible universe, with each case being mutually exclusive. ■

Figure 7 graphically shows the three cases in Theorem 4.

Corollary 1. *Based on Theorem 4, the lower bound of η for convergence to the system optimum equals to zero, if and only if the system optimum is a Nash equilibrium. Otherwise, the lower bound must be larger than zero for convergence to the system optimum.*

In summary, with effective RL algorithms, the proposed framework must converge to an optimal CE that has no larger total travel time than NE. The additional reward provided to CAVs has a positive effect in reducing congestion. As the additional reward increases, the learning can achieve a better solution with less expected total travel time and even to the SO.

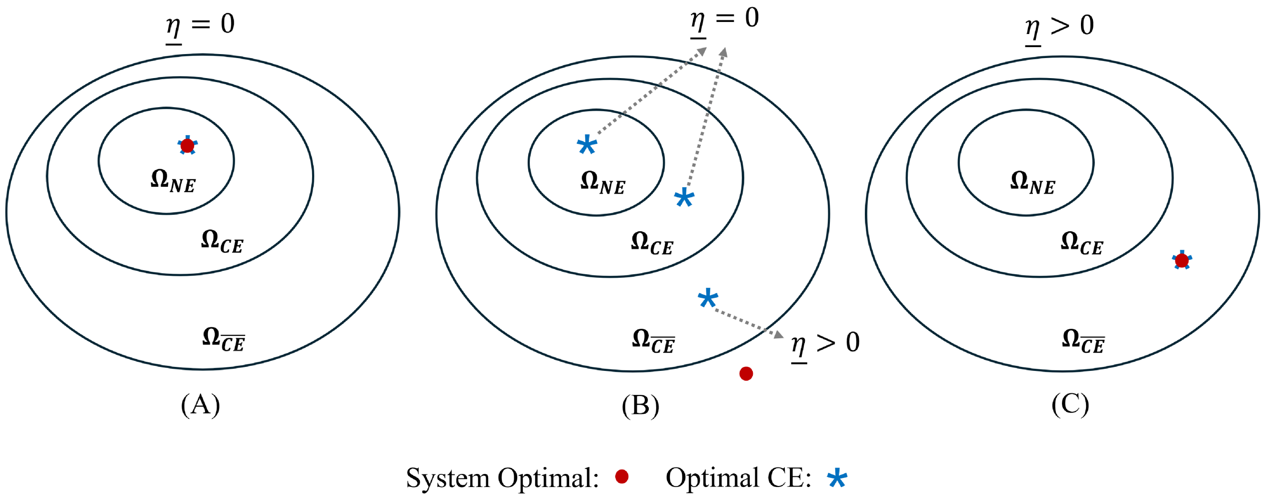


Figure 7: Demonstration of the three cases in Theorem 4

Remark 1. *For a specific learning task, the additional reward needed for converging to an optimal CE or SO should be larger than the lower bound derived by the theoretical analysis. The reason is as follows. The premise of the concept of CE includes the assumption that travelers believe others must follow the third party's suggestions. However, in the proposed framework, we do not enforce CAVs to follow the suggestions. Instead, they are intelligent agents that conduct learning. Therefore, the additional reward aimed at encouraging CAVs to follow the suggestions is needed to help meet the premise of CE. And the value of the additional reward that can make it effective can vary based on many factors, including the specific RL algorithms, the problem itself, and other hyperparameters. A small additional reward being ineffective may even lead to worse learning performance than the case without an additional reward since it may act as a negative perturbation or noise to the reward functions in the learning.*

The proposed research leverages RL and CAVs systems, addressing the challenges brought by the complexity and behavioral issues in achieving a CE. With the proposed framework, the expected travel times of the system and the individual travelers are both reduced. Meanwhile, the expected individual travel time is the same for all CAVs in the same OD pair, ensuring fairness. Therefore, this research indeed helps mitigate the UO-SO conflict effectively.

6 Numerical Example

In this section, we demonstrate the results of implementing the proposed two-way deep RL approach on the Braess and Nguyen-Dupuis networks, respectively.

6.1 Braess network

In the numerical example on the Braess network, we continue to use the same setting as in Section 5.1 for the sake of validation and comparison. The difference is that we use our proposed two-way RL approach herein, which enables en-route learning and en-route information provision, in contrast to the static case in Section 5.1. Figure 8 shows the learning results for the two cases with and without route suggestions from the information provider. The total travel times are smoothed by applying a moving average to every 50 adjacent episodes.

We first discuss the results for the case without the information provider. In this case, the CAVs make decisions based on the time, location, and real-time travel times of sub-routes, without any suggestions. The Markov game will converge to NE, since each agent will find their optimal policy that is the best response to other agents' policies (Littman, 2001; Chen et al., 2022). For the example of the Braess network in this paper, in addition to the UE, there is another NE, or more precisely, a weak NE. The route flow pattern is $[4,2,4]$, and the resulting route travel time pattern is $[8,7,7]$, and the total travel time is 74. This is a weak NE, since all travelers cannot unilaterally change their behavior to get better off. However, this is not a UE, since the travel times on each route are not equal. In this case, Figure 8 shows the multi-agent RL converges to the range between 74 and 80 and can oscillate inside the range, since both are NE.

If we involve an information provider conducting RL at the same time to influence the learning process of the CAVs, with the objective of minimizing the total travel time, the learning converges faster and to a much lower value of total travel time, as shown in Figure 8. Figure 9 shows the change in route flows by episode, in which the integers are smoothed by moving average. In fact, it converges to the SO for this example with an additional reward $\eta = 2$. The route flow pattern at the SO is $[5,1,4]$, the route travel time is $[8,6,6]$, and a minimized total travel time is 70. The result does not compromise the expected individual travel times compared to UE while achieving the system goal. In addition, in the figure, the small peak around the 400-th episode is resulted from the periodic exploration we used, since we restored the strength of the exploration for the learning of the information provider at the 400-th episode for this example. Section 6.3 will provide more details and discussions about the reward shaping and periodic exploration.

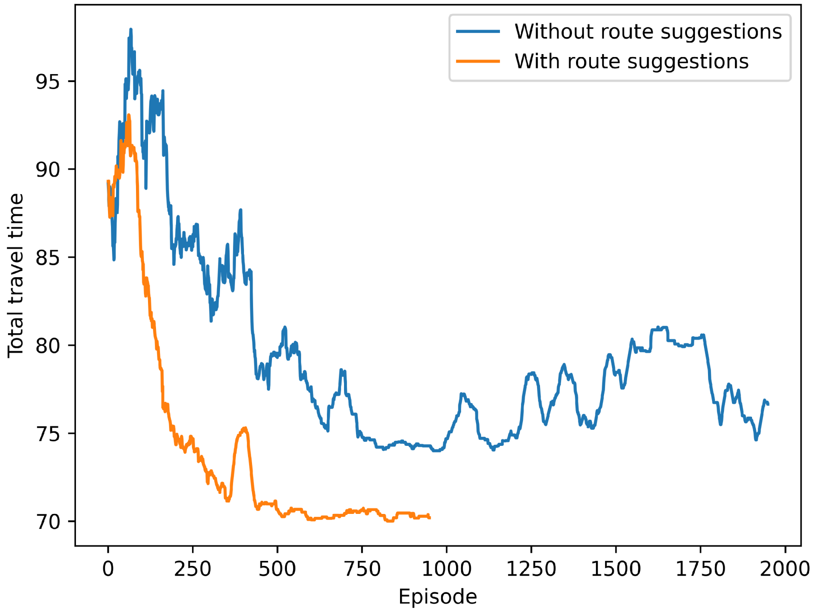


Figure 8: Total travel time by episode with and without route suggestions

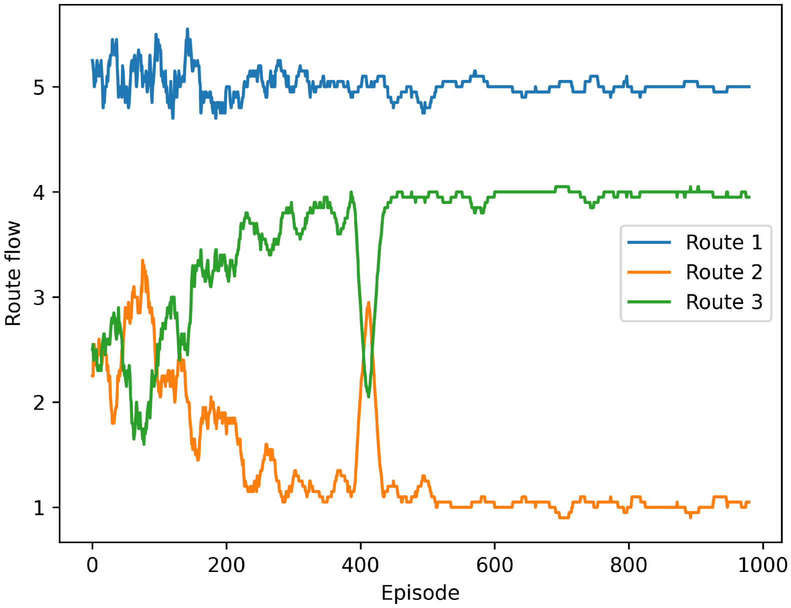


Figure 9: Route flow by episode with route suggestions

6.2 Nguyen-Dupuis network

In the Braess network example, we assume all travelers depart at the same time traveling for a single OD pair. Now, we use the Nguyen-Dupuis network, shown in Figure 10, to demonstrate the case where 960 CAVs depart during a 30-minute time horizon, traveling between four OD pairs. In the learning process, the episode ends when all travelers finish their trips.

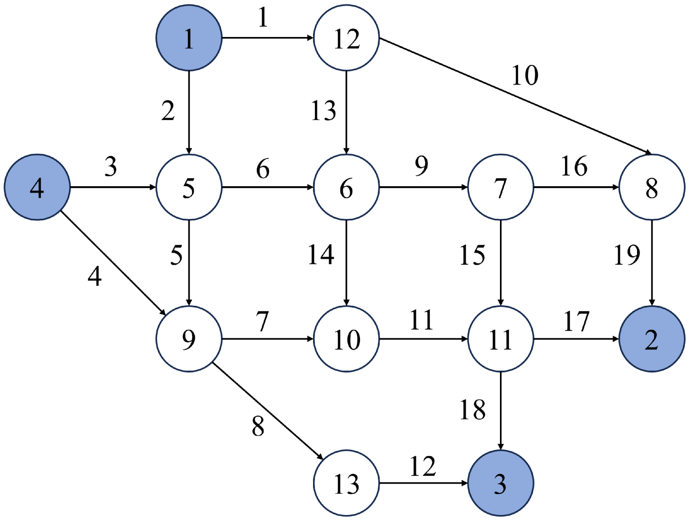


Figure 10: Nguyen-Dupuis network

This network includes two origins (nodes 1 and 4) and two destinations (nodes 2 and 3), resulting in a total of four OD pairs. The OD demand is assumed to be 480, 420, 540, and 480 veh/hour for the OD pair (1,2), (1,3), (4,2), and (4,3), respectively. The link characteristics are used as the same in Wang et al. (2023). The link travel times are computed using triangular fundamental diagrams. The demand of each OD is assumed to be evenly distributed in the 30-minute horizon for departure.

Due to the limited computation resource, it is too expensive to maintain separate DNNs for every single CAV. Therefore, in this example, we divide the CAVs into groups in which the CAVs in the same group share the same experience buffer and the same DNNs so that the total amount of training work can be reduced. Specifically, we divide the CAVs into three groups for each OD pair. Note that each CAV still takes action individually, which refers to decentralized execution under CTDE. In practice, the individual CAV can conduct learning locally if supported with advanced technologies such as edge computing.

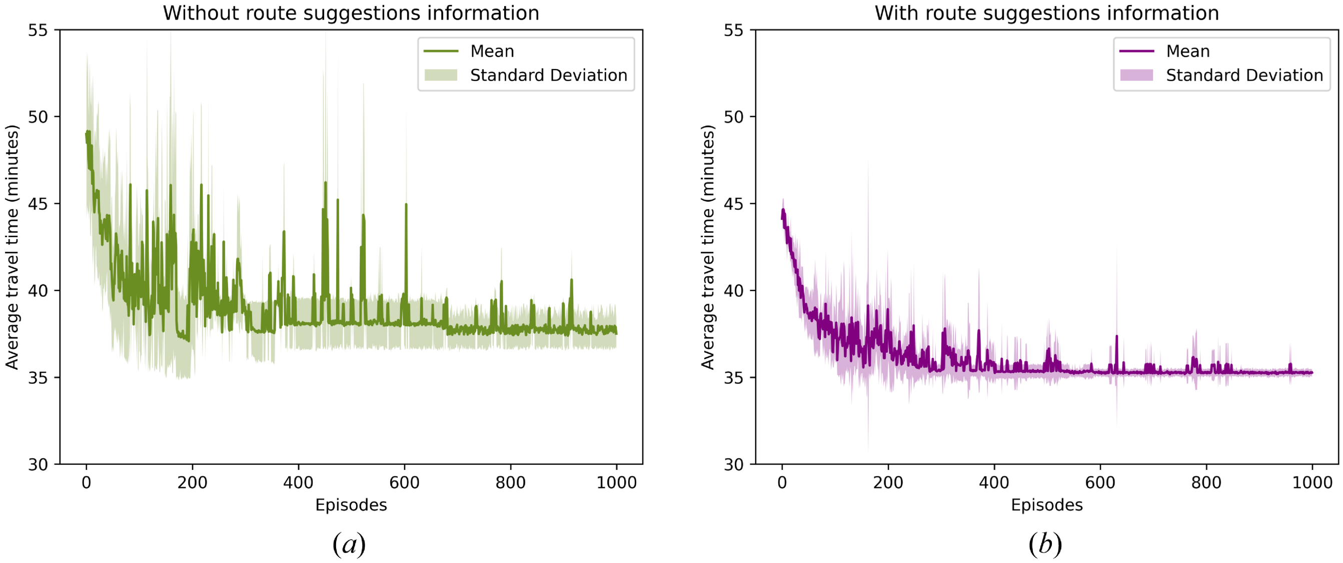


Figure 11: Average travel time by episode on the Nguyen-Dupuis network

Figures 11(a) and (b) show the results of change in average travel time per CAV during the learning process for the cases with and without en-route suggestions. For each case, we ran the simulation ten times. The center line is the mean value, and the shadow part represents the standard deviation. As we can see, with the information on en-route suggestions, the average travel time can converge to a lower value compared to the case without such information.

6.3 Discussion

This subsection provides further discussions on the reward shaping and periodic exploration techniques employed in numerical experiments.

First, we introduce how the additional reward is designed. As presented, an additional reward will be given to CAVs that follow the suggestion at a decision node. The total additional reward for following suggestions all the time, i.e., following a complete-route suggestion at the end, must be the same for all CAVs in all OD pairs, avoiding fairness concerns. To achieve this, we first compute the base units of rewards allocated for different suggestions at each decision node for all routes in all OD pairs in the network. The additional reward for following a specific suggestion at a decision node is calculated by multiplying the total additional reward η by the base unit. The base units for the Braess and Nguyen-Dupuis networks are computed as shown in Figure 12 and Figure 13, respectively. One can verify that the total unit for following any complete route in any OD pair is equal to 1.

In each episode, if the CAV follows the received suggestion every time at a decision node, then it will receive the total additional reward η. If it follows the suggestions only at some of the decision nodes, it will receive a portion of the additional rewards, depending on the decision nodes where it follows the suggestions. These additional rewards are added upon the negative travel times in the reward function of CAVs during the learning.

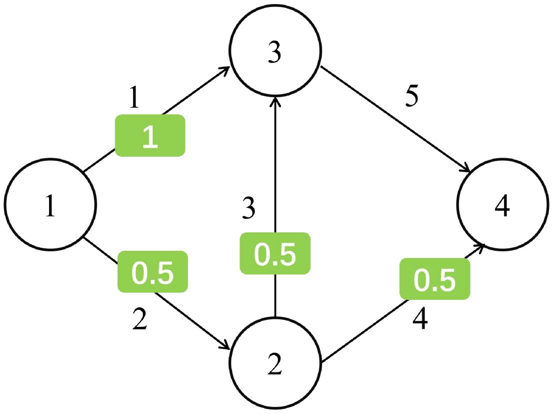


Figure 12: Base units for reward shaping in Braess network

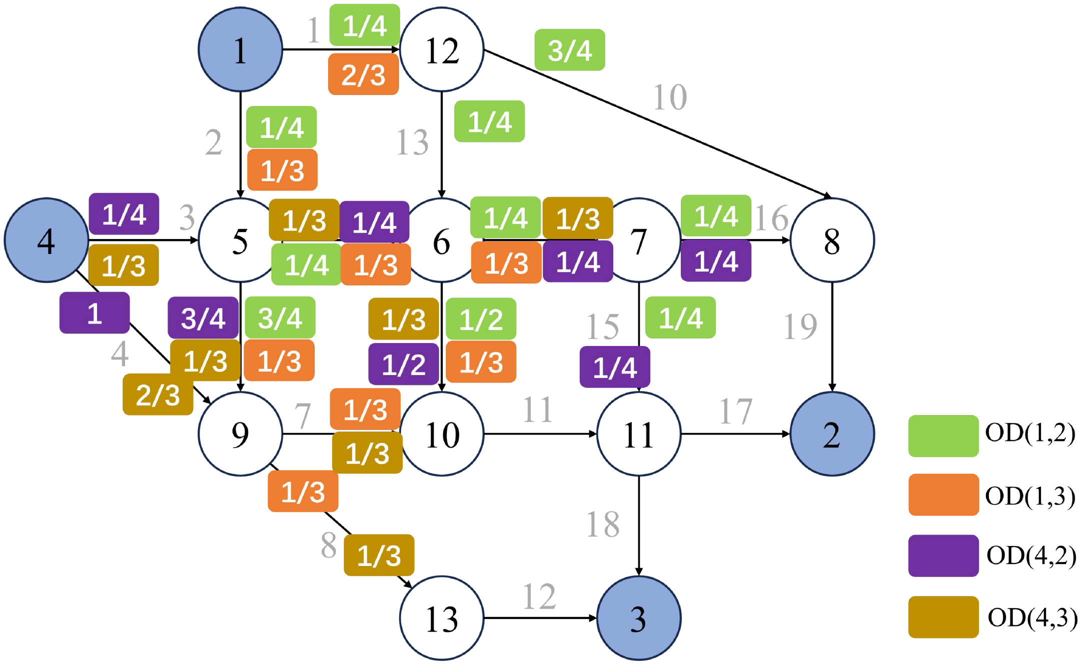


Figure 13: Base units for reward shaping in Nguyen-Dupuis network

Assume the value of travel time of CAVs is $0.1 per minute. Figure 14 shows the learning results on the Braess network where additional reward of $1, $2, and $3 is designated for following suggestions, respectively. All the three cases shown here do not use any periodic exploration.

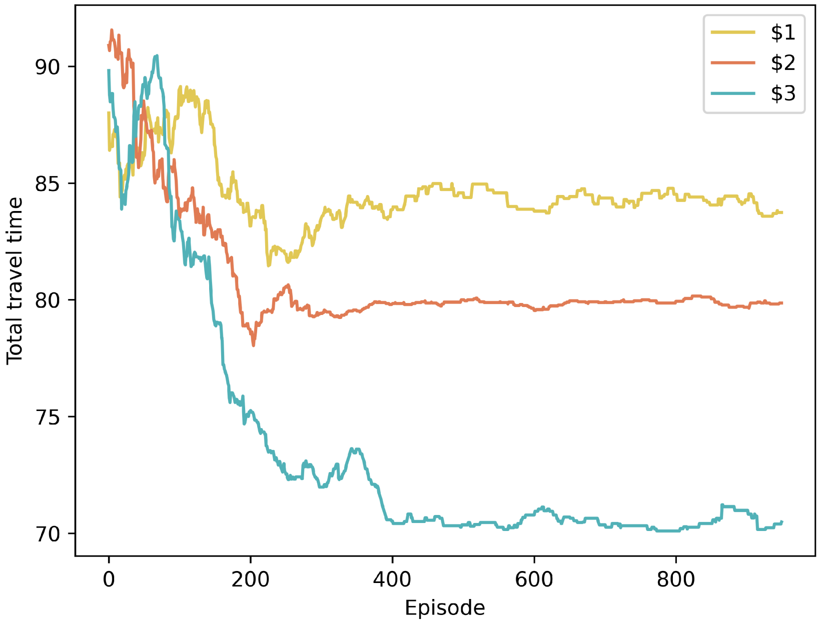


Figure 14: Effect of different additional rewards on the Braess network

The results in Figure 14 suggest that higher additional rewards can facilitate a better learning process, consistent with Theorem 3. Also, as discussed before in Section 5.3, when the additional reward is small, it may worsen the learning compared with the case (shown in Figure 8) where there is no information provider providing en-route suggestions. This is because that if the additional reward is not effective enough to encourage CAVs to follow the suggestions, it may exert a negative impact on the learning since it messes with the travel time in CAVs' reward function and reduces the learning effectiveness of Q-values for CAVs and thus for the information provider that is learning from the CAVs as well.

Another finding is that when the additional reward is large ($3 in this example), it does not need the periodic exploration to help with the learning, compared with the case where $2 is set as the additional reward and periodic exploration is employed in Figure 8. This implies that, a higher additional reward can enhance the learning effectiveness of the information provider at an earlier stage of the learning, where its exploration has not been significantly reduced. However, it is important to consider financial factors when considering increasing the additional reward. Therefore, it is recommended to judiciously use periodic exploration together as needed in practice.

Figure 15 shows the learning results on Nguyen-Dupuis network with the additional reward set at $2, $4, and $6 for following a complete route, respectively. Similar findings can be found in the figure. That is, higher additional rewards can facilitate a better learning process and result. In the results in Figure 11(b), $6 is used.

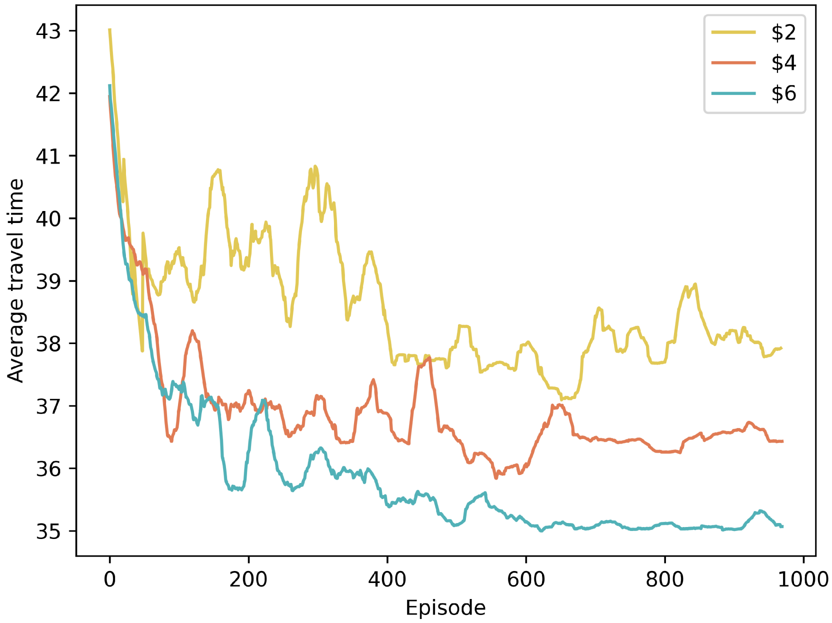


Figure 15: Effect of different additional rewards on Nguyen-Dupuis network

Note that the reward shaping here essentially differs from traditional congestion pricing/incentives schemes. In traditional congestion pricing/incentives schemes, pricing serves as a penalty for travelers using faster routes while incentives serve as compensation for travelers using slower routes. There is indeed compromise in UO in terms of travel time. However, in this paper, the reward shaping is used to help the two-way RL between CAVs and the information provider. Unlike traditional schemes, the reward shaping here does not sacrifice or compensate CAVs in terms of travel time, because the learning fosters correlation among CAVs so that their expected travel time are the same. The information provider provides suggestions based on its learned probability distributions, and the CAVs may be suggested different routes from day to day but their expected travel time does not violate UO. Moreover, the CAVs get the same amount of additional reward for following suggestions.

Next, we explain how the periodic exploration on the information provider side is used in the experiments. The periodic exploration is not always used but serves as an auxiliary tool to tune the learning process when specific learning behaviors suggest a need for further exploration. For example, in the experiments for the Braess network, in the case where setting $2 as the additional reward (the middle curve in Figure 14), the total travel time stops decreasing after the 200th episode (moving average is used). This observation motivates us to enhance the exploration after the 200th episode. Figure 16 shows the results for the cases where no periodic exploration, enhanced exploration at the 200th, and the 400th episodes, respectively. This figure shows that improper adjustment of the exploration may worsen the learning (e.g., at the 200th episode). The learning trajectory can be different at different runs, so the exploration adjustment can be quite uncertain. Hence, it is used as an auxiliary tool for tuning. We did not use any periodic exploration in the experiments for the Nguyen-Dupuis network.

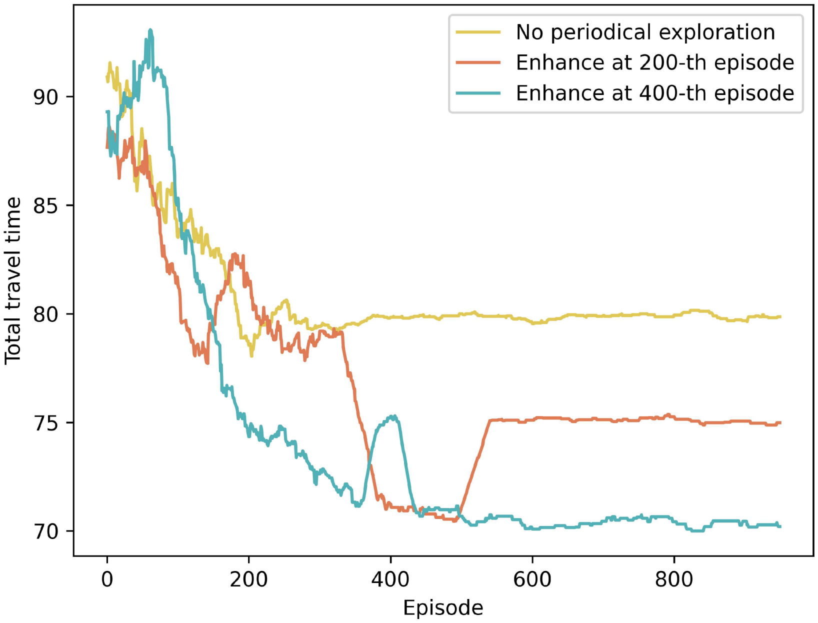


Figure 16: Experiments of periodic exploration on Braess network

7 Conclusions

This research presents a pioneering endeavor to harness the advantages of correlated equilibrium in CAV systems, and demonstrate the potential for mitigating the conflict between system-level and individual-level goals through information provision. By formulating the en-route routing behavior of CAVs as a multi-agent Markov game and the information provision mechanism as a single-agent Markov decision process, we have designed a two-way deep reinforcement learning approach for CAVs and the information provider. Our customized framework captures the unique features of the Markov game of CAVs and the information provider's MDP. Numerical examples demonstrate the effectiveness of the proposed framework for congestion mitigation without compromising individual optimality.

The key insight of this research lies in the role of the self-learning information provider, who learns the optimal en-route suggestions for CAVs through repeated interactions with the system. By providing correlated en-route suggestions that benefit both the system and individual CAVs, the information provider establishes a win-win situation. Concurrently, CAVs repeatedly explore the environment and learn optimal routing policies under the guidance of information provision, surpassing what they could learn independently without any correlation. Indeed, the proposed research leverages RL and CAVs systems to achieve CE. Theoretical analysis of the CE in the proposed framework provides insights and a foundation for mitigating the UO-SO conflict.

This research opens up several promising avenues for more sophisticated and effective approaches to tackling the challenges in future intelligent transportation systems. First, expanding the objective of individual CAVs beyond travel time by including other relevant factors is an intriguing research direction. Second, considering potential heterogeneity among CAVs would provide more flexibility and insights for enhancing the overall CAV system performance. Finally, investigating the implications of multiple information providers in CAV systems would be an interesting area for further research. It will yield valuable insights by investigating cooperative or competitive relationships among these information providers.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grant CMMI-2047793. The authors are grateful to the Center for Infrastructure, Transportation, and the Environment at RPI for its support of this research. Additionally, we extend our appreciation to the three anonymous reviewers for their constructive comments and suggestions. The authors are solely responsible for the contents of this paper.

References

Adkins, R. P., Mount, D. M., and Zhang, A. A. (2019). *A Game-Theoretic Approach for Minimizing Delays in Autonomous Intersections*. Springer International Publishing.

Aumann, R. J. (1987). Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55.

Bazzan, A. L. and Grunitzki, R. (2016). A multiagent reinforcement learning approach to en-route trip building. *Proceedings of the International Joint Conference on Neural Networks*.

Bell, M. G. (2000). A game theory approach to measuring the performance reliability of transport networks. *Transportation Research Part B: Methodological*, 34(6):533–545.

Bellman, R. (1957). A markovian decision process. *Indiana University Mathematics Journal*, 6.

Chen, Z., Ma, S., and Zhou, Y. (2022). Finding correlated equilibrium of constrained markov game: A primal-dual approach. *NeurIPS Conference*.

Cigler, L. and Faltings, B. (2011). Reaching correlated equilibria through multi-agent learning. *10th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*.

de O. Ramos, G., Bazzan, A. L., and da Silva, B. C. (2018). Analysing the impact of travel information for minimising the regret of route choice. *Transportation Research Part C: Emerging Technologies*, 88.

Du, L., Han, L., and Chen, S. (2015). Coordinated online in-vehicle routing balancing user optimality and system optimality through information perturbation. *Transportation Research Part B: Methodological*, 79.

Gairing, M., Monien, B., and Tiemann, K. (2008). Selfish routing with incomplete information. *Theory of Computing Systems*, 42.

Gronauer, S. and Diepold, K. (2022). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55.

Grunitzki, R., Ramos, G. O. D., and Bazzan, A. L. C. (2014). Individual versus difference rewards on reinforcement learning for route choice. *Proceedings - 2014 Brazilian Conference on Intelligent Systems, BRACIS*.

Jo, Y., Jang, J., Ko, J., and Oh, C. (2022). An in-vehicle warning information provision strategy for v2v-based proactive traffic safety management. *IEEE Transactions on Intelligent Transportation Systems*, 23.

Kim, H., Kim, W., Kim, J., Lee, S.-J., and Yoon, D. (2019). A study on the effects of providing situation awareness information for the control authority transition of automated vehicle. *International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1394–1396.

Ko, J., Kim, H., Oh, C., and Kim, S. (2023). Impact of v2v warning information on traffic stream performance using microscopic simulation based on real-world connected vehicle driving behavior. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14.

Liang, X., Guler, S. I., and Gayah, V. V. (2023). Decentralized arterial traffic signal optimization with connected vehicle information. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 27.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Machine Learning Proceedings*.

Littman, M. L. (2001). Value-function reinforcement learning in markov games. *Cognitive Systems Research*, 2.

Liu, Y. and Whinston, A. B. (2019). Efficient real-time routing for autonomous vehicles through bayes correlated equilibrium: An information design framework. *Information Economics and Policy*, 47.

Ma, J., Smith, B. L., and Zhou, X. (2016). Personalized real-time traffic information provision: Agent-based optimization model and solution framework. *Transportation Research Part C: Emerging Technologies*, 64:164–182.

Mao, W. and Başar, T. (2022). Provably efficient reinforcement learning in decentralized general-sum markov games. *Dynamic Games and Applications*.

Marris, L., Muller, P., Lanctot, M., Tuyls, K., and Graepel, T. (2021). Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. *International Conference on Machine Learning*, pages 7480–7491.

Ning, Y. and Du, L. (2023). Robust and resilient equilibrium routing mechanism for traffic congestion mitigation built upon correlated equilibrium and distributed optimization. *Transportation Research Part B: Methodological*, 168:170–205.

Paz, A. and Peeta, S. (2009). Behavior-consistent real-time traffic routing under information provision. *Transportation Research Part C: Emerging Technologies*, 17(6):642–661.

Puterman, M. L. (2008). Markov decision processes: Discrete stochastic dynamic programming. *Wiley Series in Probability and Statistics*.

Shou, Z., Chen, X., Fu, Y., and Di, X. (2022). Multi-agent reinforcement learning for markov routing games: A new modeling paradigm for dynamic traffic assignment. *Transportation Research Part C: Emerging Technologies*, 137.

Spana, S. and Du, L. (2022). Optimal information perturbation for traffic congestion mitigation: Gaussian process regression and optimization. *Transportation Research Part C: Emerging Technologies*, 138.

Spana, S., Du, L., and Yin, Y. (2022). Strategic information perturbation for an online in-vehicle coordinated routing mechanism for connected vehicles under mixed-strategy congestion game. *IEEE Transactions on Intelligent Transportation Systems*, 23.

Stefanello, F., Silva, B. C. D., and Bazzan, A. L. (2016). Using topological statistics to bias and accelerate route choice: Preliminary findings in synthetic and real-world road networks. *CEUR Workshop Proceedings*, 1678.

Su, H., Zhong, Y. D., Chow, J. Y., Dey, B., and Jin, L. (2023). Emvlight: A multi-agent reinforcement learning framework for an emergency vehicle decentralized routing and traffic signal control system. *Transportation Research Part C: Emerging Technologies*, 146:103955.

Tony, L. A., Ghose, D., and Chakravarthy, A. (2022). Correlated-equilibrium-based unmanned aerial vehicle conflict resolution. *Journal of Aerospace Information Systems*, 19:283–304.

Wang, C., Peeta, S., and Wang, J. (2021). Incentive-based decentralized routing for connected and autonomous vehicles using information propagation. *Transportation Research Part B: Methodological*, 149.

Wang, L., Zhao, L., Hu, X., Zhao, X., and Wang, H. (2023). A reliability-based traffic equilibrium model with boundedly rational travelers considering acceptable arrival thresholds. *Sustainability*, 15:6988.

Zhang, K., Yang, Z., and Başar, T. (2021). *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. Springer.

Zhou, A., Wang, J., and Peeta, S. (2022). Robust control strategy for platoon of connected and autonomous vehicles considering falsified information injected through communication links. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*.

Zhou, B., Song, Q., Zhao, Z., and Liu, T. (2020). A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. *Applied Mathematics and Computation*, 371:124895.