This is the Pre-Published Version.

This is an Accepted Manuscript of an article published by Taylor & Francis in Transportmetrica B: Transport Dynamics on 02 Mar 2023 (published online), available at: http://www.tandfonline.com/10.1080/21680566.2023.2179461.

A Bibliometric Analysis and Review on Reinforcement Learning for Transportation Applications

Can Li[a], Lei Bai[b], Lina Yao[a], S. Travis Waller[c], Wei Liu[d,*]

[a] *School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia*
[b] *School of Electrical and Information Engineering, University of Sydney, Sydney, NSW 2008, Australia*
[c] *Lighthouse Professorship "Transport Modelling and Simulation", Faculty of Transport and Traffic Sciences, Technische Universität Dresden, Germany*
[d] *Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China*

Abstract

Transportation is the backbone of the economy and urban development. Improving the efficiency, sustainability, resilience, and intelligence of transportation systems is critical and also challenging. The constantly changing traffic conditions, the uncertain influence of external factors (e.g., weather, accidents), and the interactions among multiple travel modes and multi-type flows result in the dynamic and stochastic natures of transportation systems. The planning, operation, and control of transportation systems require flexible and adaptable strategies in order to deal with uncertainty, non-linearity, variability, and high complexity. In this context, Reinforcement Learning (RL) that enables autonomous decision-makers to interact with the complex environment, learn from the experiences, and select optimal actions has been rapidly emerging as one of the most useful approaches for smart transportation applications. This paper conducts a bibliometric analysis to identify the development of RL-based methods for transportation applications, representative journals/conferences, and leading topics in recent ten years. Then, this paper presents a comprehensive literature review on applications of RL in transportation based on the specific topics. The potential future research directions of RL applications and developments are also discussed.

Keywords: Machine Learning; Reinforcement Leaning; Transportation; Bibliometric Analysis

1. Introduction

The travel demand is increasing along with the growth of social and economic activities, which results in great challenges in terms of crowding, congestion, emission, energy, and safety. Meanwhile, a massive amount of multi-source data has been continuously and/or automatically collected. In this context, artificial intelligence (AI) methods that can take advantage of the growing data availability have been proposed to address challenges faced by transportation systems and travelers and thus improve system safety, sustainability, resilience, and efficiency.

Reinforcement Learning (RL) is an essential branch of AI-based methods, which is an experience-driven autonomous learning strategy for decision-making that aims to obtain the maximum accumulative reward. The concepts and terminologies in relation to reinforcement learning are first proposed in 1954 (Minsky, 1954), where the trial and error interaction with the environment is emphasized as the core mechanism of RL to learn optimal behaviors/decisions (Kaelbling et al., 1996). Bellman (1957) proposes the dynamic programming method to solve the discrete Markov Decision Process (MDP) for the optimal control problem, where the proposed method is similar to the trial and error mechanism, and thus MDP becomes the most common mathematical framework to define RL tasks. Later on, Q-learning is proposed (Watkins, 1989) to find the optimal strategy under limited information/knowledge (e.g., without the knowledge of the state transition

*Corresponding author
Email address: wei.w.liu@polyu.edu.hk (Wei Liu)

function), which further expands the application of RL. Since the development of Q-learning, applications with RL have grown rapidly. For instance, RL algorithms have been applied for Atari games proposed by DeepMind (Mnih et al., 2015). The design of AlphaGo (Silver et al., 2016), a deep RL-based Go program, defeats advanced human players, which demonstrates the huge potential of deep reinforcement learning.

In the past several years, many top conference papers and journal papers have reported diverse theoretical progress of RL, which have motivated wide applications of RL in different fields. For instance, RL-based methods are able to control complex machinery (Levine et al., 2016) and self-driving (Wang et al., 2019). Also, it has been applied in recommendation systems for commodity recommendation (Chen et al., 2018) and advertising placement (Lou et al., 2020). The utilization of RL in the natural language processing (NLP) domain has also been explored extensively, such as dialogue system (Mo et al., 2018) and context sequence modeling (Chen et al., 2021). In addition, RL can be used to improve communication network resource allocation efficiency (Mao et al., 2016), where the energy usage for data centers can be reduced.[1] The wide applications of Reinforcement Learning in different domains demonstrate the advantages of RL, which are further explained below. First, RL does not necessarily require substantial prior experiences or historical data to train the agent (Ye et al., 2019). Second, model-free RL algorithms allow agents to learn the environment information for optimization without dependence on prior expert knowledge. Third, RL is able to handle long-term problems by acknowledging long-term returns rather than only considering an immediate return for short-term benefits (Pan et al., 2019). Also, multi-agent RL algorithms that can handle large-scale systems where multiple agents either cooperate or compete with each other have been proposed. Multi-agent RL shows strong scalability by distributing tasks appropriately for a large number of agents (Desjardins and Chaib-Draa, 2011).

In line with the advantages of RL, many studies have developed and/or applied RL strategies in the transportation sector. The experimental results evaluated on real-world datasets or synthetic datasets demonstrate the effectiveness of Reinforcement Learning in learning and managing transportation systems, improving accuracy and efficiency, and reducing resource consumption. There are several existing reviews on RL studies in the transportation domain. In particular, Mannion et al. (2016); Yau et al. (2017); Noaeen et al. (2022) focus on traffic signal control with RL; Aradi (2022); Kiran et al. (2022); Zhu and Zhao (2021) focus on deep RL models for autonomous driving; and Qin et al. (2022) focuses on RL algorithms for ride-sharing. Three additional review studies (Abdulhai and Kattan, 2003; Haydari and Yilmaz, 2022; Farazi et al., 2021) have covered more transportation applications with Reinforcement Learning. Abdulhai and Kattan (2003) is published in 2003, which does not cover the substantial development of RL methods in recent years. Farazi et al. (2021) mainly focuses on deep RL methods for applications in transportation (e.g., autonomous driving and traffic signal control). However, non-deep RL models have not been examined. Haydari and Yilmaz (2022) has discussed both deep RL and non-deep RL methods and covers a wide range of RL applications in transportation (including traffic signal control, energy management for the electric vehicle, road control, and autonomous driving). However, the importance of fairness in developing RL methods for transportation applications is not emphasized. Moreover, none has provided a bibliometric analysis of RL methods for transportation applications. Differently, this study takes advantage of the bibliometric analysis to provide a systematic review on applications of both deep RL and non-deep RL methods in transportation, and provide more comprehensive coverage of applications than related existing reviews (e.g., including RL applications in taxi and bus systems that have not been covered by Haydari and Yilmaz (2022)). Besides, this paper further points out several aspects that require substantial efforts in terms of developing RL methods for real-world transportation applications, i.e., scalability, practicality, transferability, and fairness.

Specifically, this study provides a summary on applications of RL to address relevant transportation issues and takes advantage of the bibliometric analysis approach to uncover connections among the journals/conferences and use keywords to identify the influential journals/conferences and areas of concern. Several future directions of RL studies in transportation are also discussed. The major transportation topics that involve RL methods discussed in this study include traffic

[1] https://www.deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40

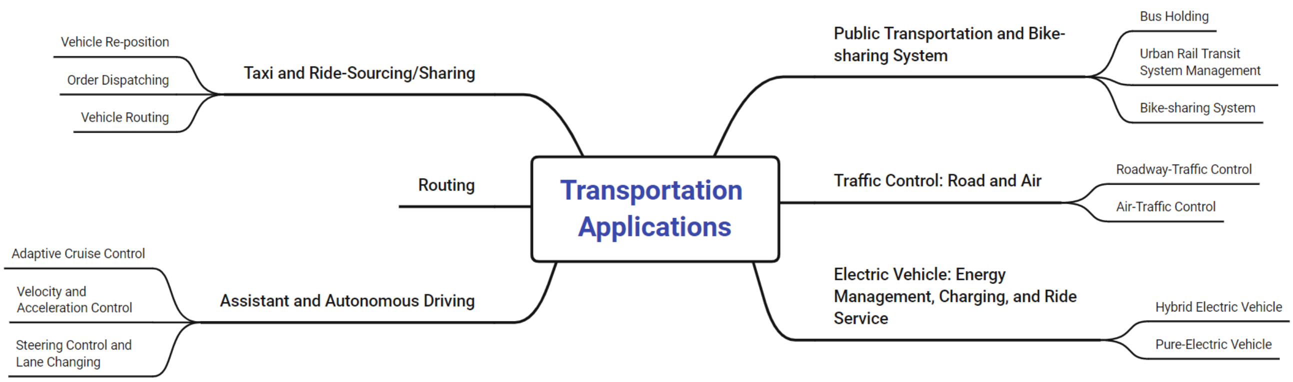


Fig. 1. Classification of RL Applications in Transportation

control, taxi and ride-sourcing/sharing, assistant and autonomous driving, routing, public transportation and bike-sharing system, and electric vehicles, which are identified based on an analysis of keywords summarized in Section 3. The detailed classification of topics is shown in Fig. 1. In particular, this review has collected over six hundred related papers mostly published in the last thirteen years in major journals in the transportation domain (e.g., Transportation Research Part B, Part C, IEEE Transactions on Intelligent Transportation Systems, IET Intelligent Transport Systems) and major related conferences in the computer science domain (e.g., AAAI, KDD, WWW, CIKM), which will be further discussed in Section 3. To summarize, this paper provides a reference point to researchers for interdisciplinary Reinforcement Learning research in transportation and computer science.

The rest of this paper is structured as follows. Section 2 introduces basic formulations of Reinforcement Learning and Section 3 conducts the bibliometric study. The review of the six topic categories for transportation applications with RL are presented in Section 4 – Section 9, respectively. Future directions of RL in transportation and the conclusion of this paper are discussed in Section 10.

2. Preliminary

Markov Decision Process (MDP) is often used to provide the basic mathematical formulation for Reinforcement Learning, which is presented first in this section. Then, algorithms for Reinforcement Learning (including value-based algorithms, policy-based algorithms, and actor-critic-based algorithms) and data usage in transportation applications are discussed.

2.1. Markov Decision Process

MDP is a mathematical model for stochastic control processes that can simulate agents, stochastic policy, and rewards, which provides a mathematical framework for RL (Sutton and Barto, 2018). RL aims to maximize the reward where the MDP framework is able to produce the delayed reward by adopting the reward function and discount factor. In MDP, the Markov property is a fundamental concept, which is defined as the next state being only related to the current state and is independent of previous states (Markov, 1954). The Markov property (state independence) often helps simplify the optimization task of RL.

In detail, MDP consists of five elements, i.e., $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$, where $\mathcal{S}$ represents the set of states, $\mathcal{A}$ denotes the set of actions, $\mathcal{P}$ is the probabilistic transition function, $\mathcal{R}$ is the reward function, and $\gamma \in [0, 1]$ denotes the discount factor. At time step t, under a state $s_t \in \mathcal{S}$, the agent performs an action $a_t \in \mathcal{A}$ and then receives an immediate reward $r_t(s_t, a_t) \in \mathcal{R}$ from the environment. The environment state will change to $s_{t+1} \in \mathcal{S}$ based on the transition probability $\mathcal{P}(s_{t+1}|s_t, a_t)$. The goal of the agent is to find an optimal policy π^* for maximizing the cumulative reward with a discount factor where $\mathcal{G} = \sum_{t=1}^{T} \gamma^t r_t$, $\pi^* = \text{argmax}_\pi \mathbb{E}[\mathcal{G}|\pi]$, and $\mathbb{E}$ represents the expectation operator. Specifically, the state, action, and reward are all problem-specific. For instance, for traffic signal control problems, the state may include traffic flow and speed information, the action is the signal timing, and the reward is often defined to minimize traffic delay. The transition dynamics matrix maps the pair of the state and action into the distribution of states in the next time step, which consists of the probability between any two states. The specific values of transition matrices often do not need to be calculated after the development of Q-learning. The discount factor is often adopted to put more weight on more recent return. The policy is the solution to MDP, which maps from the state to the action and indicates the action to be taken under the specific state.

Depending on the number of agents that are considered, RL can be divided into single-agent and multi-agent algorithms. When there are multiple agents, three relations among agents are often considered, i.e., the fully competitive, the semi-competitive and semi-cooperative, and the fully cooperative. Compared to single-agent RL, multi-agent RL faces more challenges. For example, the joint actions of all agents will affect the state, which increases the instability of the environment and leads to the difficulty of optimization. Also, in a multi-agent system, we may have to deal with agents with only local observation/information. In addition, the increase of agents will require more computation resources to handle the large or high-dimensional state and action spaces. This paper involves both single-agent and multi-agent RL methods for transportation applications.

2.2. Reinforcement Learning Algorithms

This subsection will introduce several major Reinforcement Learning algorithms, i.e., value-based algorithms, policy-based algorithms, and actor-critic-based algorithms, which are different in terms of how they optimize the decisions.

Different states/outcomes (in future time steps) may occur even under the same actions (at the current time step). Therefore, expected cumulative rewards are often considered. In particular, the state-value function $V^\pi(s)$ calculates the expected cumulative reward under state s and policy π. The state-action function $Q^\pi(s,a)$ calculates the expected cumulative reward of taking action a under state s. The state-value function and the state-action function can be formulated as follows:

$$V^\pi(s) = \mathbb{E}[\mathcal{G}|s] \tag{1}$$

$$Q^\pi(s,a) = \mathbb{E}[\mathcal{G}|s,a] \tag{2}$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s,a) \tag{3}$$

$$Q^\pi(s,a) = \sum_{s'} \mathcal{P}(s'|s,a)(r(s,a) + V^\pi(s')) \tag{4}$$

Then, the optimal policy is obtained by letting $\pi(s) = \text{argmax}_a Q(s,a)$ and the state-value function is $V^\pi(s) = \max_a Q^\pi(s,a)$. Bellman Expectation Equation (Bellman, 1952) can be used to solve the value function:

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V^\pi(s')] \tag{5}$$

With the above value functions, one then aims to produce the optimal policy that maximizes the long-term reward, where dynamic programming is often used to solve the problem, based on value iteration, policy iteration, or their combination. For the value iteration approach (value-based RL), after the initialization for the state-value function, there are two major steps (to be repeated), i.e., (i) calculating the state-action value for each pair of the action and state and (ii) updating the value function by choosing the maximum state-action value as the current state value. The above two steps will be repeated until the state-value function convergence. For the policy iteration approach (policy-based RL), after selecting an initial policy, there are two main steps (to be repeated), i.e., (i) policy evaluation by the state-value function and (ii) calculating the best action under the current state for policy improvement. The policy evaluation and policy improvement are repeated continuously until the policy no longer changes. Actor-Critic-based RL combines value-based and policy-based approaches. The above three strategies based on value iteration, policy iteration, or their combination are introduced below.

2.2.1. Value-based Reinforcement Learning

In value-based RL, the value function $V^{\pi}(s)$ is updated following the Bellman Optimal Equation (Bellman, 1952) and Eq. (5) can be rewritten as:

$$V_{k+1}^{\pi}(s) = \max_{a} \mathbb{E}[r_{t+1} + \gamma V_k^{\pi}(S_{t+1})|(S_t = s, A_t = a)] \tag{6}$$

Two classic approaches have been used to estimate $V^{\pi}(s)$, i.e., Monte-Carlo-based approach (MC) and Temporal-Difference-based approach (TD). In MC, based on current state $s(t)$, the agent starts to interact with the environment until reaching a termination condition. Then, the cumulative reward $\mathcal{G}_t$ can be calculated. The value-based RL tries to drive $V_t^{\pi}(s)$ close to $\mathcal{G}_t$, which updates the value-function as follows:

$$V_t^{\pi}(s) \leftarrow V_t^{\pi}(s) + \alpha(\mathcal{G}_t - V_t^{\pi}(s)) \tag{7}$$

where α is the learning rate. Since the reward obtained by MC is estimated at the end of the episode in concern, there can be large variances in the cumulative reward. On the contrary, TD only simulates one step in the episode in concern and updates the value-function as follows:

$$V_t^{\pi}(s) \leftarrow V_t^{\pi}(s) + \alpha(r_t + \gamma V_t^{\pi}(s+1) - V_t^{\pi}(s)) \tag{8}$$

which yields smaller variances but can be less accurate due to a lack of a systematic consideration of the whole episode.

Typical TD-based strategies are Q-learning (Watkins and Dayan, 1992) and State-Action-Reward-State-Action (Sarsa) algorithm (Sutton, 1996), which replace $V^{\pi}(s)$ with $Q^{\pi}(s,a)$ following Eq. (8). The update policy of Q-learning can be expressed as:

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha(r_t + \gamma max_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)) \tag{9}$$

And the update policy of Sarsa can be expressed as:

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha(r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)) \tag{10}$$

Both Q-learning and Sarsa involve (i) a behavior policy to interact with the environment and sample potential actions from the learning data with randomness and (ii) a target policy to improve the performance with the help of sampling data and thus obtain the optimal policy. The "off-policy method" updates the target policy based on the data generated from the behavior policy, while the "on-policy method" updates the target policy based on the data generated by itself (Sutton et al., 1998). Sarsa is an on-policy method (i.e., the target policy is the same as the behavior policy), while Q-learning is an off-policy method (i.e., the target policy is to suppose the selecting action with the largest reward to update the value function).

Q-learning might not be able to accommodate a large number of states and actions in some applications. Therefore, different deep models have been embedded in Q-learning to approximate the value function to deal with such issues. Mnih et al. (2015) proposes Deep Q-Network (DQN) for optimal policy finding. Given a Q-function Q and a target Q-function $\hat{Q}$ initialized as $\hat{Q} = Q$, an experience replay buffer is utilized to store the transition (s_t, a_t, r_t, s_{t+1}) in each time step where a_t is obtained by Q. When enough sample data is obtained from trials with the environment, a mini-batch of samples is randomly selected to produce the target y (a target point that provides the direction to move in order to improve the solution) as follows:

$$y = r_t + \gamma \max_{a} \hat{Q}(s_{t+1}, a) \tag{11}$$

Then, parameters of Q are updated by driving $Q(s_t, a_t)$ towards y with the gradient descent method. The target network $\hat{Q}$ will be reset by $\hat{Q} = Q$ after a number of C steps, where the value of C is a hyper-parameter to decide the iteration step for updating the parameters of the target network. It is noteworthy that for the combination of deep learning and RL two issues remain. The samples (in the aforementioned to produce y in Eq. (11)) to be generated when combining deep learning with RL are independent, while the states often have correlations. Moreover, the distribution of targets is static in deep learning, but the states are continuously varying in RL. Thus, the experience replay buffer designed in DQN is used to accommodate the non-static distribution problem and correlations of states. Furthermore, the instability problem caused by the usage of non-linear neural networks to represent value functions can be solved by properly designing the target network. Moreover, the $\epsilon - greedy$ strategy is often used to increase randomness when generating actions to balance exploration and exploitation.

Further DQN-based methods such as Double-DQN (Van Hasselt et al., 2016) and Dueling-DQN (Wang et al., 2016) are developed for more robust and faster policy learning. In detail, to reduce the overestimations caused by the single estimator of Q-learning (i.e., the estimated value is larger than the true value) (Thrun and Schwartz, 1993), Double Q-learning implements the choice and the evaluation of actions with double-estimator where two Q-functions are defined, i.e., $Q^A(s,a)$ and $Q^B(s,a)$ (Van Hasselt, 2010). Specifically, each Q-function is updated with the value obtained from the other Q-function in the next state, which can be expressed as follows:

$$\begin{aligned} Q^A(s_t,a_t) &\leftarrow Q^A(s_t,a_t) + \alpha(r_t + \gamma max_{a_{t+1}} Q^B(s_{t+1}, argmax_a Q^A(s_{t+1},a_t)) - Q^A(s_t,a_t)) \\ Q^B(s_t,a_t) &\leftarrow Q^B(s_t,a_t) + \alpha(r_t + \gamma max_{a_{t+1}} Q^A(s_{t+1}, argmax_a Q^B(s_{t+1},a_t)) - Q^B(s_t,a_t)) \end{aligned} \tag{12}$$

Van Hasselt et al. (2016) further embeds deep learning into Double Q-learning and proposes Double-DQN. The evaluation of the current policy is estimated by the target network $\hat{Q}$ instead of the second network in Double Q-learning. And the derivation of the target y in Double-DQN is obtained as follows:

$$y = r_t + \gamma \hat{Q}(s_{t+1}, argmax_a Q(s_{t+1}, a)) \tag{13}$$

Similar to the target network in DQN, the target network in Double-DQN keeps fixed and updates after a predetermined number of steps by $\hat{Q} = Q$.

Dueling-DQN replaces the output state-action value function of DQN by the combination of the state-value function and the advantage function, i.e., $Q^\pi(s_t,a_t) = V^\pi(s_t) + A^\pi(s_t,a_t)$, where $A^\pi(s_t,a_t)$ is the advantage function for the strategy evaluation. The design of the advantage function helps identify whether rewards are mainly an outcome of the state or induced by different actions. The suitability of specific actions can be evaluated.

Given the success of DQN for decision-making, numerous variants of DQN have been proposed. For instance, Prioritized Replay DQN (Tom et al., 2016) is designed such that important transitions are selected more frequently, and thus can help improve efficiency. Multi-step Learning (Yinlong et al., 2019) is proposed such that return in multiple steps is used instead of the reward in one step in order to reduce the bias and accelerate training. Noisy Network (Fortunato et al., 2017) approach replaces the $\epsilon - greedy$ strategy by adding noises on parameters to enhance the exploration ability. Moreover, Rainbow (Hessel et al., 2018) is proposed to combine Dueling DQN, Prioritized Replay, Multi-step Learning, Distributional RL, and Noisy Net to further improve the performance.

2.2.2. Policy-based Reinforcement Learning

Policy-based Reinforcement Learning algorithms model and estimate the policy function directly and optimize the policy function to maximize the reward. Specifically, REINFORCE (Williams, 1992) optimizes policy π_θ with the parameter vector θ by maximizing the expected return r_t where the gradient is approximated by the stochastic gradient descent technique for parameter updating. Based on REINFORCE, Sutton et al. (2000) introduces the Policy Gradient method to optimize policy $\pi_\theta(s,a)$ by maximizing the average reward $\rho(\pi) = \sum_s d^\pi(s) \sum_a \pi(s,a) r(s,a)$ as follows:

$$\frac{\partial \rho}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^\pi(s,a) \tag{14}$$

where $d^\pi(s) = lim_{t\to\infty} P(s_t = s|s_0, \pi)$ represents the stationary distribution of states under π and $Q^\pi(s,a) = \sum_{t=1}^{\infty} \mathbb{E}[r_t - \rho(\pi)|s_0 = s, a_0 = a, \pi]$. In MDP starting from a stationary state, $d^\pi(s)$ can also be defined as the discounted weighting of states under policy π starting at state s_0 and $Q^\pi(s,a) = \mathbb{E}[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}|s_t = s, a_t = a, \pi]$. Then, Q^π is approximated by an estimator f_w

and thus the Policy Gradient with Function Approximation can be written as:

$$\frac{\partial \rho}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} f_w(s,a) \tag{15}$$

where $\frac{\partial f_w(s,a)}{\partial w} = \frac{\partial \pi(s,a)}{\partial \theta} \frac{1}{\pi(s,a)}$. Thus, the gradient can be expressed in a suitable form to find the locally optimal policy.

Further policy-based algorithms are also designed. For instance, Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) is proposed, which tends to give monotonic improvement over iterations by constraining the Kullback–Leibler divergence between the old and updated policies so that the change of the entire parameter space will not be too large to avoid the collapse of state values caused by wrong decisions. Similarly, Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a widely adopted algorithm to ensure the difference between the old and updated policies is also not too large by limiting the ratio between old and updated strategies under a hyper-parameter value.

2.2.3. Actor-Critic-based Reinforcement Learning

Actor-Critic-based (AC-based) RL (Sutton et al., 2000) takes advantage of both value-based function and policy-based function. The actor network interacts with the environment and generates actions. The critic network uses the value function to evaluate the performance of the actor and guide the actor's actions in the next time step.

Some widely-used algorithms in AC-based RL are Deterministic Policy Gradient (DPG) (Silver et al., 2014), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016), Advantage Actor-Critic (A2C) (Mnih et al., 2016), and Asynchronous Advantage Actor-Critic (A3C) (Babaeizadeh et al., 2017). DPG and DDPG are off-policy methods that can be trained even in high-dimensional action space, and DDPG adopts deep learning into DPG. A2C and A3C are on-policy algorithms where A2C adopts a synchronous control method, and A3C adopts an asynchronous control method for actor network updating. A3C is often adopted in transportation problems for policy-making, which is further discussed below as an example to illustrate the mechanism of asynchronous methods. A3C takes advantage of the Actor-Critic framework and introduces the asynchronous method to improve performance and efficiency. Multiple threads are utilized in A3C to collect data in parallel, i.e., each thread is an independent agent to explore an independent environment. Also, each agent can use different strategies to sample data where sampling data independently is able to obtain unrelated samples and increase sampling speed.

2.3. Data

Synthetic and real-world data have been used in studies for transportation applications with RL. On the one hand, it is easier and more feasible to obtain synthetic data. A large number of scenarios/samples with different characteristics can be constructed to evaluate proposed methods. However, some uncertainties, disruptions, and accidents occurring in practice are hard to be measured or simulated, which leaves a certain and unknown gap with actual environments. On the other hand, the real-world data can reflect the actual situations more accurately, which means that the proposed method can be put into practice for the scenario corresponding to the collected data. It is harder to obtain complete and diverse real-world data due to several reasons, e.g., the confidentiality of various sources and the lack of information. Also, a real-world dataset may only represent the characteristics of a specific target, which has limited scenarios/samples to evaluate the generality of proposed models.

Although the applications and corresponding data are diverse, the type of data can be divided into three categories, i.e., road network relevant data, traffic flow relevant data, and vehicle operation relevant data. Specifically, road networks are regarded as directed graphs with nodes and edges (i.e., nodes denote intersections while edges represent roads). Some other road related characteristics (e.g., speed limit, the number of lanes/tracks, and distributions of bus/railway stations) are also concluded to construct the stationary environment of RL. The traffic flow relevant data (e.g., traffic speed and demand) and vehicle operation relevant data (e.g., fuel/electricity consumption, vehicle speed/acceleration, and lane changing) are used as the time-varying input of RL models to constitute the dynamic environment of RL. The agents learn and analyze the information of both stationary and dynamic environments for decision-making based on different RL-based optimization strategies.

3. Bibliometric Analysis

This section provides a bibliometric analysis of studies for RL-based transportation applications. The distribution of published papers in journals/conferences and the characteristics of research fields or topics are explored. The VOSviewer software [2] is used to measure the quantities and connections in relation to publications and keywords.

The selected journals and conferences covering January 2010 to December 2022 are summarized in Table 1 according to the number of published related papers. The list of journals and conferences is based on the following. The selected transportation-related journals are ranked as Q1, Q2, and Q3 by Scimago Journal & Country Rank in 2022.[3] The selected conferences in the field of artificial intelligence and data mining are with the highest CORE ranking (CORE A+) in recent years.[4]. International Conference on Intelligent Transportation Systems (ITSC) is also included due to its high relevance and wide audience. It can be seen that ITSC covers a substantial number of RL-based transportation applications studies (i.e., about 27.24%), which indicates that Reinforcement Learning has attracted substantial attention for achieving intelligent traffic control and management. Other journals with considerable relevant publications are T-VT, T-ITS, and TR-C with 173 (28.22%), 100 (16.31%), and 49 (7.99%) papers, respectively, which indicates the fusion and interaction of traditional transportation applications and popular machine learning strategies over the recent decade. Several transportation journals involve a relatively small number of papers regarding applications of RL (e.g., TR-A, TR-D, and Transportmetrica A), indicating that there are significant research potentials here for developing advanced RL in diverse aspects of transportation.

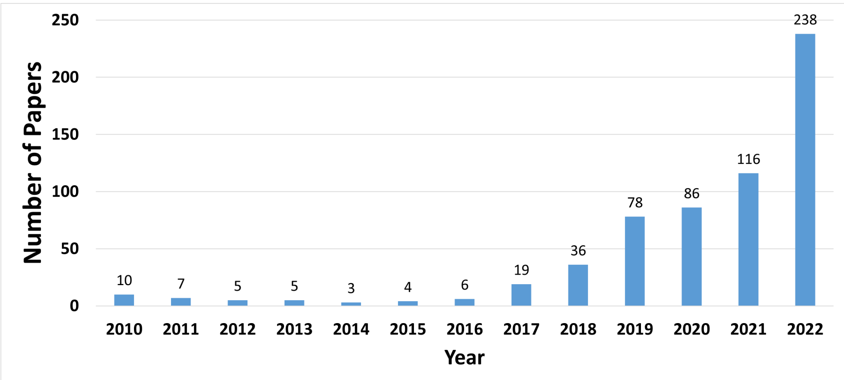


Fig. 2. Number of Published Related Papers per Year (Jan. 2010 - Dec. 2022)

In addition, the numbers of the published papers in the aforementioned journals and conferences from January 2010 to December 2022 are shown in Fig. 2. Before 2017, only a few studies per year focused on Reinforcement Learning to solve transportation problems, with only 40 articles published in total in the selected journals and conferences. And the number of published related papers from 2011 to 2016 is between three and seven (around five), which is regarded as a random fluctuation. In the following six years (i.e., 2017-2022), the number of related papers has grown substantially, which indicates the increasing importance and popularity of RL to deal with transportation problems.

Furthermore, in order to identify the major transportation application areas/topics in relation to Reinforcement Learning, Fig. 3 shows the bibliographic coupling network of keywords where the minimum number of occurrences of a keyword is five. The size of the circle represents the

[2] https://www.vosviewer.com/
[3] https://www.scimagojr.com/journalrank.php
[4] http://cic.tju.edu.cn/faculty/zhileiliu/doc/COREComputerScienceConferenceRankings.html

Table 1
Numbers of Related Publications in Major Journals/Conferences (as of December 31, 2022)

Attribute	**Name**	**Number of Related Papers**
Journal	IEEE Transactions on Vehicular Technology (T-VT)	173
Conference	IEEE International Conference on Intelligent Transportation Systems (ITSC)	167
Journal	IEEE Transactions on Intelligent Transportation Systems (T-ITS)	100
Journal	Transportation Research Part C: Emerging Technologies (TR-C)	49
Journal	IET Intelligent Transport Systems	19
Journal	IEEE Transactions on Transportation Electrification	16
Conference	Association for the Advancement of Artificial Intelligence (AAAI)	15
Conference	Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)	13
Journal	Transportation Research Record: Journal of the Transportation Research Board	12
Journal	Transportation Research Part E: Logistics and Transportation Review (TR-E)	10
Conference	International Joint Conference on Artificial Intelligence (IJCAI)	8
Conference	Conference on Information and Knowledge Management (CIKM)	8
Journal	Transportation Research Part B: Methodological (TR-B)	5
Journal	Transportmetrica B: Transport Dynamics	4
Conference	World Wide Web Conference (WWW)	4
Conference	International Conference on Data Mining (ICDM)	3
Journal	Transportation	1
Journal	Transportation Science	1
Journal	Transportation Research Part F: Traffic Psychology and Behaviour (TR-F)	1
Journal	Journal of Transportation Engineering Part A: Systems	1
Journal	Research in Transportation Economics	1
Journal	Journal of Air Transport Management	1
Journal	Travel Behaviour and Society	1
Journal	Transport Reviews	0
Journal	Transportation Research Part A: Policy and Practice (TR-A)	0
Journal	Transportation Research Part D: Transport and Environment (TR-D)	0
Journal	Journal of Transport Geography	0
Journal	Transportmetrica A: Transport Science	0
Journal	Transport Policy	0
Journal	International Journal of Sustainable Transportation	0
Journal	Maritime Policy & Management	0
Journal	Journal of Transportation Engineering, Part B: Pavements	0

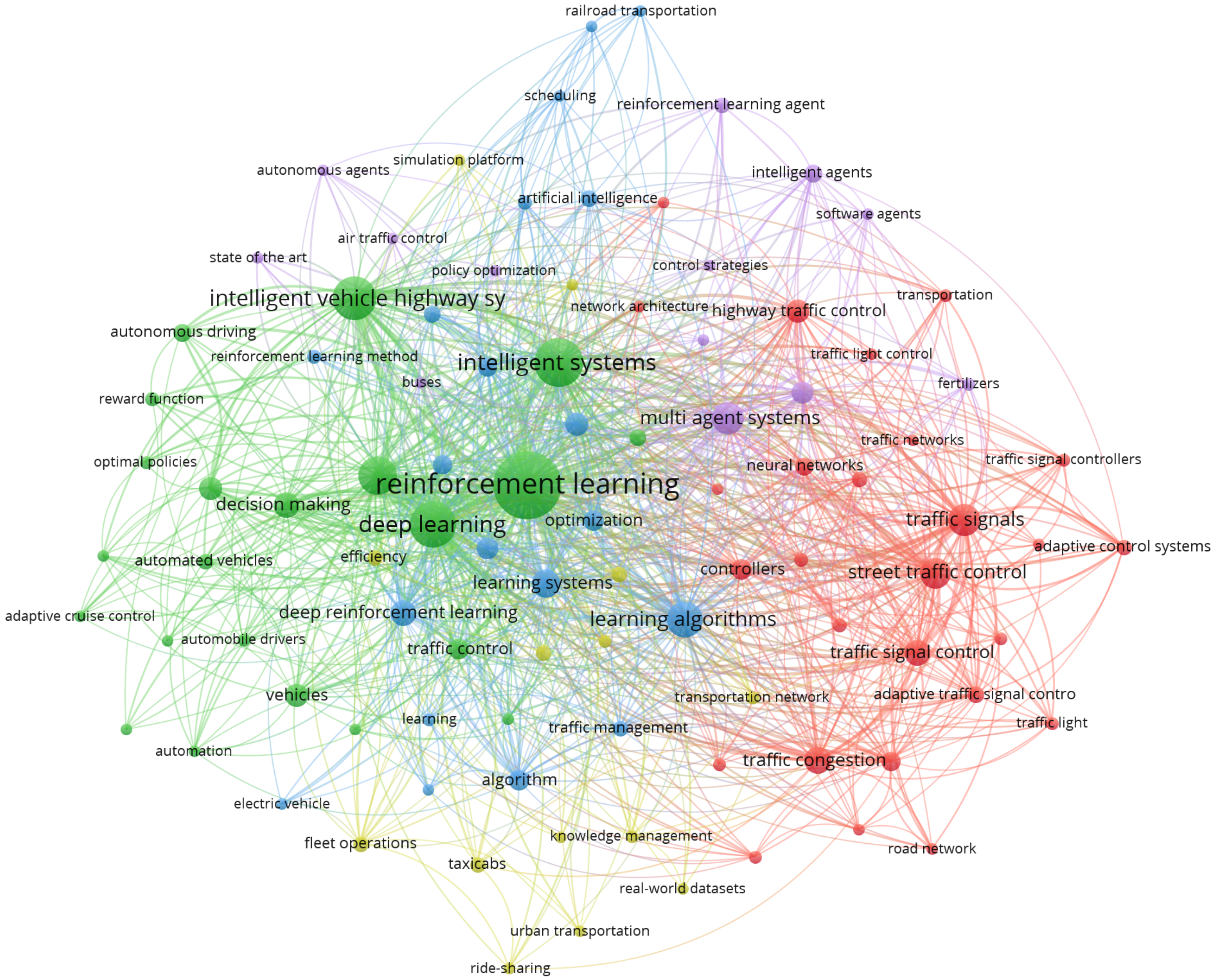


Fig. 3. Bibliographic Coupling of Keywords: the circle represents a keyword while the edge represents the co-appearance of a pair of keywords.

number of occurrences of the keyword. And the keywords represented by the same color mean the high co-appearance of these words in one paper. Excluding the words with similar meanings, the keywords with high frequency can be described as two aspects, i.e., learning algorithms and intelligent transportation applications. The learning strategies mainly cover deep learning or Neural Network and Reinforcement Learning. The major topics related to RL methods include the following nine categories: autonomous driving/vehicles, adaptive cruise control, fleet operations, ride-sharing, traffic signal control, highway/street/air traffic control, electric vehicle, taxicabs, and scheduling. Motivated by these keywords with high frequency, we identify six groups as shown in Fig. 1, which will be reviewed in the following sections, respectively.

4. Traffic Control: Road and Air

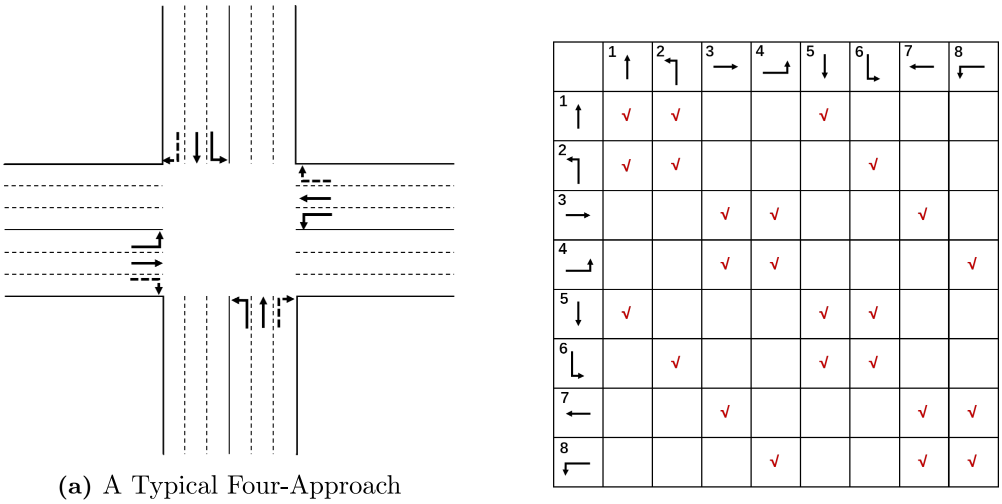
Traffic control is a critical issue in traffic flow management. This section summarizes RL-based controlling strategies proposed for both roadway traffic and air traffic in order to reduce traffic congestion and delays. Due to the large number of studies for traffic signal control and to facilitate reading, we summarize studies on roadway traffic signal control (TSC) in Table 2 and summarize studies on other aspects (i.e., speed limit, price management, perimeter control, and air traffic control) in Table 3.

4.1. Roadway Traffic Control

On roadway traffic control, we review the following five major issues: traffic signal control; speed limit control; pricing management; perimeter control; and ramp metering.

4.1.1. Traffic Signal Control

The congestion and delays caused by traffic bottlenecks motivate the development of methods for traffic signal control (TSC) (Yau et al., 2017). Conventional pre-timed control systems set constant time signals, while RL-based approaches have been used to dynamically and adaptively optimize traffic signal timing. We first illustrate a four-approach intersection as depicted in Fig. 4a (left-hand driving is assumed) and a typical signal plan with eight phases as shown in Fig. 4b. Many studies are formulated based on the such four-approach intersections with eight phases (Arel et al., 2010).



(a) A Typical Four-Approach Intersection

(b) Phase Competition Matrix

Fig. 4. Traffic Signal Related Schematic Diagrams

The studies for traffic signal control started with the exploration for one intersection with single-agent RL methods, which provides the fundamental methods for TSC in environments with multiple intersections. Specifically, for one intersection, an intuitive designing scheme is to regard the intersection as an agent for signal control policy optimization and the agent's decision is subject

Table 2
Summary of RL Applications in Traffic Signal Control

		Reference
Framework	**Q-learning**	Prashanth and Bhatnagar (2010), Ozan et al. (2015), El-Tantawy et al. (2013, 2014), Mannion et al. (2015), Reyad and Sayed (2022), Wiering (2000), Balaji et al. (2010), Arel et al. (2010), Abdoos et al. (2011)
	DQN	Mousavi et al. (2017), Wei et al. (2018), Zhang et al. (2020a), Xu et al. (2019), Van der Pol and Oliehoek (2016), Darmoul et al. (2017), Devailly et al. (2021), Wang et al. (2021a), Wei et al. (2018, 2019a,b), Chen et al. (2020), Zang et al. (2020), Zhang et al. (2021b), Yu et al. (2020), Xu et al. (2021)
	A2C	Chu et al. (2019), Wang et al. (2021a)
	DDPG	Li et al. (2021b), Ni and Cassidy (2019)
	Actor-Critic	Aslani et al. (2017)
	Neural fitted Q-iteration	Nishi et al. (2018)
	Ape-X DQN	Zheng et al. (2019)
Agent	**single-agent**	Prashanth and Bhatnagar (2010), Ozan et al. (2015), Reyad and Sayed (2022), El-Tantawy et al. (2014), Mousavi et al. (2017), Xu et al. (2019), Wei et al. (2018), Zhang et al. (2021b), Ni and Cassidy (2019)
	multi-agent	Nishi et al. (2018), Wiering (2000), Abdulhai et al. (2003), Abdoos et al. (2011), Chu et al. (2019), Balaji et al. (2010), El-Tantawy et al. (2013), Arel et al. (2010), Van der Pol and Oliehoek (2016), Yu et al. (2020), Wang et al. (2021a), Zheng et al. (2019), Chen et al. (2020), Xu et al. (2021) Devailly et al. (2021), Mannion et al. (2015), Zang et al. (2020), Zhang et al. (2020a), Wei et al. (2019a,b), Li et al. (2021b), Darmoul et al. (2017), Aslani et al. (2017)
Scenario/ Data	**synthetic network/data**	Prashanth and Bhatnagar (2010), Abdoos et al. (2011), Ozan et al. (2015), El-Tantawy et al. (2014), Mousavi et al. (2017), Nishi et al. (2018), Wiering (2000), Abdulhai et al. (2003), Arel et al. (2010), Van der Pol and Oliehoek (2016), Darmoul et al. (2017), Reyad and Sayed (2022), Mannion et al. (2015), Aslani et al. (2017), Ni and Cassidy (2019)
	real-world network/data	Wei et al. (2018) (Jinan), Zheng et al. (2019) (Jinan, Hangzhou), Zhang et al. (2020a) (Hangzhou, Atlanta), Chu et al. (2019) (Monaco), Wang et al. (2021a) (Monaco, Harbin), El-Tantawy et al. (2013) (Toronto), Li et al. (2021b) (Maryland), Zang et al. (2020) (Jinan, Hangzhou, Atlanta, Los Angeles), Chen et al. (2020); Devailly et al. (2021) (New York), Balaji et al. (2010) (Singapore), Xu et al. (2019) (Hangzhou), Wei et al. (2019a) (Jinan, New York), Zhang et al. (2020d), Wei et al. (2019b); Yu et al. (2020) (Hangzhou, Jinan, New York), Xu et al. (2021) (Hangzhou, Jinan, Shenzhen, New York)
Simulator	**GLD simulator** (Wiering et al., 2004)	Prashanth and Bhatnagar (2010)
	Paramics	El-Tantawy et al. (2013, 2014), Balaji et al. (2010)
	SUMO (Lopez et al., 2018)	Mousavi et al. (2017), Wei et al. (2018), Nishi et al. (2018), Chu et al. (2019), Mannion et al. (2015), Van der Pol and Oliehoek (2016), Wang et al. (2021a), Li et al. (2021b), Devailly et al. (2021), Zhang et al. (2021b), Yu et al. (2020), Xu et al. (2019)
	CityFlow (Zhang et al., 2019a)	Zhang et al. (2020a), Wei et al. (2019a,b), Zheng et al. (2019), Chen et al. (2020), Zang et al. (2020), Yu et al. (2020), Xu et al. (2021)
	AIMSUN [1]	Aslani et al. (2017), Ni and Cassidy (2019)
	VISSIM [2]	Darmoul et al. (2017), Reyad and Sayed (2022)
	personal simulator	Ozan et al. (2015), Wiering (2000), Abdoos et al. (2011), Abdulhai et al. (2003), Arel et al. (2010)

[1] http://www.AIMSUN.com
[2] http://vision-traffic.ptvgroup.com/en-uk/home

Table 3
Summary of RL Applications in Speed Limit Control, Price Management, Perimeter Control, and Air Traffic Control

Reference	Application	Framework	Agent	Scenario/Data	Simulator
Zhu and Ukkusuri (2014)	speed limit control	TD-based RL	single-agent, the controller	Sioux Falls network	personal simulator
Li et al. (2017b)	speed limit control	Q-learning	single-agent, the controller	Interstate freeway in Oakland	personal simulator
Wu et al. (2020b)	speed limit control	DDPG	single-agent, the controller	northbound freeway of I405 in California	SUMO
Pandey and Boyles (2018)	price management	Sparse Cooperative Q-learning	multi-agent, a toll	synthetic network	personal simulator
Pandey et al. (2020)	price management	A2C, PPO	multi-agent, a toll	express lanes in Dallas and Austin	personal simulator
Zhou and Gayah (2021a)	perimeter control	DQN, DDPG	single-agent, the controller	synthetic network	personal simulator
Chen et al. (2022)	perimeter control	Policy iteration	single-agent, the controller	synthetic network	SUMO
Yang et al. (2017)	perimeter control	DQN	single-agent, the controller	synthetic network	personal simulator
Rezaee et al. (2012)	ramp metering	Q-learning	single-agent, the controller	in the City of Toronto	Paramics
Fares and Gomaa (2014)	ramp metering	Q-learning	single-agent, the controller	synthetic network	personal simulator
Belletti et al. (2017)	ramp metering	DDPG	multi-agent, the controller for a region	San Francisco Bay Bridge	BeATs [1]
Tumer and Agogino (2007)	air traffic management	Q-learning	multi-agent, a location	synthetic network	FACET [2]
Balakrishna et al. (2010)	flight delay	Q-learning	single-agent, the controller	Tampa International Airport	personal simulator

[1] https://connected-corridors.berkeley.edu/berkeley-advanced-traffic-simulator
[2] https://www.nasa.gov/centers/ames/research/lifeonearth/lifeonearth-facet.html

to the setting of phases. To deal with the single-agent (one intersection) scenario, Q-learning (Prashanth and Bhatnagar, 2010; Ozan et al., 2015; El-Tantawy et al., 2014; Reyad and Sayed, 2022) and DQN (Mousavi et al., 2017; Wei et al., 2018; Zhang et al., 2021b) have been the most commonly used framework to learn the action-value function in order to reduce the total/average delay of vehicles. The deep model, DQN, for traffic light optimization is able to accommodate more complex and non-linear environmental information of an intersection. Different types of states might be adopted. For example, the congestion level (low, medium, or high) indicated by the queue lengths and elapsed times of each signaled lane (Prashanth and Bhatnagar, 2010) are designed to reduce the dimensionality of the state. Exact values regarding traffic conditions (e.g., link flows and the free-flow travel time) (El-Tantawy et al., 2014; Ozan et al., 2015), a vector of row pixel values (Mousavi et al., 2017), and the image representation of vehicles' positions (Wei et al., 2018) are collected to provide more completed environments.

The control strategies for one intersection can hardly relieve the traffic congestion in large metropolis with complex and dense networks, which motivates traffic control studies to simultaneously consider multiple intersections. As multiple intersections (especially neighboring intersections) may interact with each other, the optimal policy strategies should be considered at the target-area level to further improve traffic efficiency. Different reward functions have been used for TSC problems, i.e., the overall waiting time (Wiering, 2000; Nishi et al., 2018), overall delay (Abdulhai et al., 2003; Balaji et al., 2010) of all vehicles in multiple intersections, and the pressure (Varaiya, 2013) of all intersections (Wei et al., 2019a). Though these studies achieve satisfactory performance, the relations or impacts among various intersections have not been explored explicitly.

A series of studies focus on the coordination or competition among multiple agents/intersections to find area-wide or system-wide TSC strategies. Similar states and reward functions as aforementioned studies have been used based on various RL algorithms. Specifically, El-Tantawy et al. (2013) adopts the principle of Multi-agent Modular Q-learning (Ono and Fukumoto, 1996) to explicitly analyze the correlations of the target agent and one of its neighbor intersections to learn the joint policy. Arel et al. (2010) designs two types of agents for collaboration, a central agent extracting the information from itself and neighboring intersections to learn a value function and assist an outbound agent to schedule its own signals where Q-learning is used as the optimizing strategy. Furthermore, based on the Advantage Actor-Critic (A2C) framework, Chu et al. (2019) constructs the state of the agent as the composition of its observation and neighbor policies to achieve agents' coordination. The performance of the discussed coordination-based methods is superior to the isolated intersection models in terms of average intersection delay, queue length, link stop time, and link travel time.

In the aforementioned approaches, the agent of an intersection communicates with its adjacent locations but does not coordinate with further away intersections. A number of RL-based strategies are proposed to address more general system-wide or area-wide signal control issues. For instance, Van der Pol and Oliehoek (2016) combines multiple local Q-functions linearly as a global Q-function and utilizes the max-plus coordination algorithm (Kok and Vlassis, 2005) to optimize the joint action for multiple intersections in an area. Similarly, Mannion et al. (2015) defines Master and Slave agents where the Master agent uses a shared experience pool to deal with experiences from Master Agents for coordination. Yu et al. (2020) designs an active cross-agent communication mechanism to generate coordinated actions and uses the predicted traffic of the whole road network to mitigate the unnecessary impact of other agents' actions. Moreover, in Wang et al. (2021a), the Mobile Edge Computing server with a fixed number of Road Side Units collects and deals with the local states from target intersections. The processed information is sent back to each individual agent to decide the phase of the traffic light. Li et al. (2021b) proposes a shared knowledge container to store the information obtained from the whole environment by embedding the observation vectors through Gated Recurrent Unit (GRU). Each agent then chooses relevant features from the container to make its own decision based on the Deep Deterministic Policy Gradient (DDPG) algorithm.

The aforementioned studies test their approaches on small-scale environments for illustration (e.g., one intersection or dozens of intersections) while leaving scalability issues and large-scale applications for further research. In practice, megalopolis usually involves thousands of traffic light intersections, which has to be controlled simultaneously. In this context, some studies (Wei et al., 2019b; Zheng et al., 2019; Chen et al., 2020; Xu et al., 2021) focus on handling large-scale TSC problems based on various RL frameworks. In detail, Wei et al. (2019b) designs a graph attentional network named PressLight for agents' coordination by calculating and normalizing the importance score (i.e., the value to evaluate the importance of the information from the source intersection when determining the policy for the target intersection) for all intersections in pairs. The influence affected by relevant intersections is modeled by the combination of the representation obtained by the target agent and its corresponding importance score. However, determining the importance score in pair still occupies a large number of computation resources. To reduce the exploration space, Zheng et al. (2019) proposes the FRAP (i.e., Flipping and Rotation and considers All Phase configurations) model to calculate the phase score. The score of the target phase is obtained by the element-wise multiplication of the phase pair demand representation and the phase competition mask. The representation is obtained by the number of vehicles and the current signal phase, and the mask is derived from the phase competition matrix shown in Fig. 4b. The phase with the highest score is chosen to be the action. The in-variance to symmetries (e.g., flipping and rotation) in traffic signal control is achieved by pair-wise phase completion modeling to reduce the exploration space under complex scenarios. The method is combined with both value-based and policy-based RL algorithms for optimization. Furthermore, Chen et al. (2020) combines PressLight (Wei et al., 2019a) for reward function designing and FRAP (Zheng et al., 2019) for a faster training process with parameter sharing among the agents. The model is evaluated on a simulated environment with thousands of intersections to show its effectiveness. More recently, Xu et al. (2021) illustrates that minimizing the queue length, waiting time, or delay is not equivalent to minimizing average travel time, which motivates the design of different agents with different optimizing sub-targets (e.g., queue length). A high-level policy is then proposed to align all sub-policies and avoid directly minimizing average travel time.

The optimization for large-scale environments needs numerous computational resources and time, which limits such strategies to be put into practice. Therefore, given that insufficient relevant data or computing resources in the target area, Xu et al. (2019); Zang et al. (2020); Zhang et al. (2020a); Devailly et al. (2021) propose to transfer and adapt experiences learned from existing scenarios to new scenarios, which can reduce the reliance on sufficient data and decrease training consumption. As for the transfer strategies, Xu et al. (2019) selects the similar source and target intersections by calculating similarity values, Zang et al. (2020); Zhang et al. (2020a) adopt Meta-Reinforcement Learning (Finn and Levine, 2018), while Devailly et al. (2021) applies zero-short transfer learning (Higgins et al., 2017) into the TSC framework. As for the framework of Reinforcement Learning, Zang et al. (2020) develops a model based on FRAP (Zheng et al., 2019) and Xu et al. (2019); Zhang et al. (2020a); Devailly et al. (2021) utilize DQN directly.

The aforementioned studies focus on regular traffic situations while Darmoul et al. (2017); Aslani et al. (2017) focus on finding optimal solutions for traffic disruptions that are also practical and useful. In detail, Darmoul et al. (2017) investigates the impact of accidents on traffic light control by mitigating the concepts of primary and secondary immune responses (i.e., the disturbance on the road is regarded as an antigen and the associated control decision is denoted as an antibody). The multi-agent DQN method has been used for policy optimization. More specifically, the studied traffic network in Aslani et al. (2017) considers impatient pedestrians with illegal crossing behavior, vehicles parking beside the streets, and incidents (e.g., vehicle breakdown). The Actor-Critic framework is adopted to determine the duration of each phase (red/green light), which shows the capability of reducing average travel time when traffic disruptions have occurred. Furthermore, cordon control to determine the traffic signal metering rates is also an efficient way for vehicle inflows restriction. To find the optimal distribution for the metered vertices of roads, Ni and Cassidy (2019) adopts the Graph Convolution Network (GCN) to formulate the directed graph representation of the environment (i.e., the street network's geometry) and traffic (i.e., traffic conditions and directions of movements) of an intersection. The optimal actions are obtained via the DDPG method to maximize the metered flow passing through the cordon.

The promising performance of RL on traffic signal control problems motivates applications of RL in other transportation problems and also provides application examples.

4.1.2. Speed Limit Control

For flow maximization, speed limit control (adjusting the speed limit) is often used to drive the freeway recurrent traffic bottleneck density to be close to the desired density and thus avoid capacity drops (Liu et al., 2015b). The mechanism of conventional feedback-based strategies requires significant time (Li et al., 2017b), which stimulates adopting RL-based methods to deal with highly dynamic traffic situations in a timely manner.

The speed limit controller is often designed as the agent with various RL frameworks, where the research has evolved from discrete state formulations to continuous state formulations in order to accommodate complex and varying environments. Specifically, Zhu and Ukkusuri (2014) defines four congestion levels (i.e., free flow state, slight congestion state, moderate congestion state, and heavy congestion state) as the input state based on the flow density and optimizes the policy by the temporal difference (TD) algorithm. However, four discrete congestion levels might not be sufficient to fully depict the complicated and varying environment that would affect decision-making. Thus, Li et al. (2017b) uses the density at the downstream of the merge area, the density at the upstream mainline section, and the density on the ramp by specific variables instead of congestion levels to minimize the travel time. The posted speed limits set as integer multiples of five mph for freeway bottlenecks are determined by the Q-learning strategy. Similar state representations are utilized in Wu et al. (2020b) for variable speed limits control based on the optimization by the DDPG algorithm with single-agent. The proposed method is able to reduce congestion, accidents, and emissions by defining the reward function as the combination of total travel time, average velocity reported by detectors, the number of emergency braking vehicles, and related gas emissions. Though the research for speed limit control with RL does not receive much attention, the success of existing studies provides a solid foundation for future optimization.

4.1.3. Pricing

Dynamic pricing for managed lanes can be used to offer a premium service and alleviate congestion (Devarasetty et al., 2014). Pandey and Boyles (2018) and Pandey et al. (2020) examine pricing management via Reinforcement Learning to find optimal policies that maximize the revenue of the managed lanes. In these strategies, the vector containing the number of vehicles detected by the loop detectors is used as the state while the toll is set as the agent at the entrance of each managed line to decide the real-time price. A sparse cooperative Q-learning algorithm (Kok and Vlassis, 2006) is adopted in Pandey and Boyles (2018) while A2C and PPO are used in Pandey et al. (2020) to optimize the pricing policy.

4.1.4. Perimeter Control

Perimeter control is regarded as an efficient way for regional traffic control to optimize the network level traffic performance (Yang et al., 2017). The appealing performance obtained by RL-based optimizing strategies for traffic signal control illustrates their ability to handle complex and varying road environments. Similar environments analyzing in perimeter control and traffic signal control provide a novel direction for perimeter control, i.e., RL-based methods. Specifically, in Yoon et al. (2020), the agent determines green time ratios as discrete values with the optimization by DQN. However, this method is only able to handle discrete actions, which is less practical. To avoid relying on the full knowledge of the road network and design continuous action, Zhou and Gayah (2021a,b) proposes an RL-based scheme for an urban network composed of two homogeneous sub-regions to improve the network throughput (i.e., the number of trips completed). Discrete-RL (D-RL) model optimized by DQN and Continuous-RL (C-RL) model optimized by DDPG are designed for discrete actions and continuous actions, respectively. Acknowledging the information of accumulations and estimated traffic demands as the state, the agent of D-RL decides the range while the agent of C-RL controls the allowable decrease/increase value of perimeter controllers (i.e., the parameter defined by the allowable portions of transfer flows) by maximizing actual portions of transfer flows. In addition, Chen et al. (2022) proposes a deep-based integral policy iteration approach to minimize the total time spent for multi-region perimeter control in a continuous manner.

4.1.5. Ramp Metering

Ramp metering takes advantage of traffic signals at freeway on-ramps to control the rate of vehicles entering the freeway. To decide passing and prohibiting phases on the freeway, the information of the numbers of vehicles in the mainstream and entering the freeway and the status of the ramp traffic signal are denoted as the state in existing studies with either single-agent or multi-agent methods. Rezaee et al. (2012) and Fares and Gomaa (2014) utilize Q-learning-based methods to minimize the total travel time of the whole network and the freeway density, respectively. The proposed models have been tested on a case study (e.g., the City of Toronto) and a synthetic network, which illustrates the effectiveness of RL-based methods in dealing with the ramp metering problem. However, the aforementioned two single-agent-based methods have limited scalability for controlling numerous intersections simultaneously. This motivates Belletti et al. (2017) to design a multi-agent DDPG framework for ramp metering. The highway vehicle density is modeled by the Partial Differential Equation to decide the incoming flow by maximizing the total observed outflow with the policy gradient algorithm. The interaction among agents is achieved by the introduction of Mutual Weight Regularization (Caruana, 1997).

4.2. Air Traffic Control

Congestion in air traffic creates substantial flight delays and limits efficiency and productivity. As reported in Balakrishna et al. (2010), one of the major factors leading to flight delays is the taxi-out delay (i.e., the time between gate push back and time of takeoff). In order to mitigate congestion in the airport, a novel way to predict the delay based on RL is proposed, which has a relatively low demand on training data for optimization when compared to classical supervised learning strategies. The agent learns the information from the environment of the aircraft and airport (e.g., the number of aircraft in the queue at the runway and the number of departure aircraft co-taxiing) to estimate the taxi-out time by minimizing the absolute value of the error between the actual taxi-out time and predicted taxi-out time. In addition, Tumer and Agogino (2007) applies multi-agent Reinforcement Learning in air traffic flow management to minimize the sum of total delay penalty and total congestion penalty for all aircraft in the system. The ground locations throughout the airspace are split into multiple individual 'fixes' (i.e., individual locations) where each 'fix' is regarded as an agent. The task of the agent is to decide the distance between the approaching aircraft and itself, which can control the rate of aircraft going through a 'fix'. The proposed method is tested on a simulation tool, FACET, developed by NASA to show its ability for congestion reduction. The effectiveness of numerous RL strategies for air traffic control still has to be tested and evaluated in future research under complex and practical scenarios.

5. Taxi and Ride-sourcing/sharing

Cooperative mobility-on-demand (MOD) systems (e.g., Uber, Lyft, and Didi Chuxing) have been spreading widely (He and Shin, 2019) and provide multiple online taxi services such as express car, ride-sharing, ride-sourcing, and traditional taxi. The real-time large-scale order information provides the opportunity to analyze demand patterns for further forecasting and management. To reduce resource utilization, decrease the waiting time, and increase profit, Reinforcement Learning has been investigated for vehicle re-positioning, order dispatching, and vehicle routing in the taxi and ride-sourcing/sharing service systems, where a summary of related papers is provided in Table 4.

5.1. Vehicle Re-positioning

The imbalance between supply and demand leads to long waiting times for passengers and time/energy loss for drivers. Re-positioning available vehicles/drivers to potential locations (e.g., locations with massive demand) is necessary to improve system efficiency and better match supply and demand. Methods requiring accurate information on a wide range of parameters or variables (e.g., customer demand and travel time) are often time-consuming (Mao et al., 2020). Therefore, RL-based methods without the need for prior knowledge are broadly utilized for vehicle re-positioning in traditional taxi and ride-sourcing/sharing systems.

In the ride-hailing system, considering the influence from all vehicles and customers, existing studies (Nguyen et al., 2017; Lin et al., 2018; Shou and Di, 2020; Mao et al., 2020) take each available vehicle (or driver) as an agent for vehicle re-position, and develop various multi-agent RL models with different reward functions. For instance, gross merchandise volume (GMV, i.e., the number of all orders served) and order response rate are set as the reward function by Lin et al. (2018) with contextual DQN and Actor-Critic frameworks. The contextual DQN model is designed for the allocation instructing to filter out invalid directions and avoid conflicting directions for agents. The contextual Actor-Critic framework is designed for explicit coordination among agents to enhance policy-making by acknowledging spatial distributions of available vehicles and orders. The influence of waiting time on passenger loss is overlooked in Lin et al. (2018), while Mao et al. (2020) further considers impatient passengers that may leave the market. The cancellation cost caused by user-specific tolerance of waiting time is regarded as one of the components of the reward function. The proposed model shows its superiority in reducing the cancellation rate and total waiting time of impatient passengers for the taxi system by the Actor-Critic framework.

As for the traditional taxi system, global information, such as the distribution of all taxis, is hard to be obtained in a short time for optimization. Thus, Shou and Di (2020) develops a taxi re-positioning method that only uses local observations from each driver/vehicle through multi-agent Mean Field Actor-Critic algorithm (Yang et al., 2018). The aim of each agent (i.e., an available vehicle/driver) is to maximize their own monetary return. To accommodate the selfishness of each agent, Bayesian optimization is adopted to design the reward function, which helps achieve a better equilibrium for the overall system.

Table 4
Summary of RL Applications in Taxi and Ride-Sourcing/Sharing Service Systems

Reference	Application	Framework	Agent	Data	Simulator
Lin et al. (2018)	vehicle re-positioning	Contextual DQN and Actor-Critic	multi-agent, an available vehicle	real data from Didi Chuxing in Chengdu	contextual simulator (Lin et al., 2018)
Shou and Di (2020)	vehicle re-positioning	Mean Field Actor-Critic algorithm	multi-agent, an available vehicle	synthetic data, real data from NYC TLC [1]	personal simulator
Nguyen et al. (2017)	vehicle re-positioning	Actor-Critic algorithm	multi-agent, an available vehicle	synthetic data, real taxi data from Singapore	personal simulator
Mao et al. (2020)	vehicle re-positioning	Deep Actor-Critic algorithm	multi-agent, an available vehicle	real data from NYC TLC [1]	personal simulator
Oda and Joe-Wong (2018)	order dispatching	Double-DQN	single-agent, dispatch center	real data from NYC TLC [1]	personal simulator
Zhou et al. (2019a)	order dispatching	DQN	multi-agent, a driver	real data from Didi Chuxing of three cities	simulator provided by Didi Chuxing
Xu et al. (2018)	order dispatching	TD-based RL	multi-agent, a driver	synthetic data, real data from Didi Chuxing	personal simulator
Li et al. (2019)	order dispatching	Actor-Critic, Mean Field RL	multi-agent, a driver	real data from Didi Chuxing	contextual simulator (Lin et al., 2018)
He and Shin (2019)	order dispatching	Double-DQN	single-agent, coordination center	real data from Uber, Yellow Taxi and Didi Chuxing	personal simulator
Wang et al. (2018)	order dispatching	Double-DQN	multi-agent, a driver	ExpressCar data from Didi Chuxing	personal simulator
Tang et al. (2019)	order dispatching	TD-based RL	multi-agent, a driver	real data from Didi Chuxing	personal simulator
Jin et al. (2019)	order dispatching and vehicle re-position	Hierarchical RL, DDPG	multi-agent, a region cell	real data from Didi Chuxing	contextual simulator (Lin et al., 2018)
Holler et al. (2019)	order dispatching and vehicle re-position	DQN, PPO	multi-agent, a driver	synthetic data, real GAIA dataset from Didi Chuxing	personal simulator
Chen et al. (2019)	order dispatching and pricing	TD-based RL	single-agent, coordination center	real data from Didi Chuxing	simulator provided by Didi Chuxing
Manchella et al. (2021)	order dispatching and goods delivery	Double-DQN	multi-agent, a vehicle	real data from New York City Taxicab	personal simulator
James et al. (2019)	vehicle routing	Deep Policy Gradient algorithm	single-agent, dispatch center	real data from Cologne	personal simulator
Zhang et al. (2020b)	vehicle routing	Deep Policy Gradient algorithm	multi-agent, a vehicle	synthetic data	personal simulator
Silva et al. (2019)	vehicle routing	Q-learning	multi-agent, a vehicle	synthetic data	personal simulator
Al-Abbasi et al. (2019)	order dispatching and vehicle routing	Double-DQN	multi-agent, a vehicle	real data of taxi from NYC TLC [1]	personal simulator

[1] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

The computational complexity of the vanilla Actor-Critic-based method is relatively high for large-scale multi-agent vehicle re-positioning, which can take a very long time for convergence and is neglected in Lin et al. (2018); Shou and Di (2020). Thus, in favor of reducing the computational complexity and speeding up the optimization process, Nguyen et al. (2017) decomposes the approximation of the action-value function over agents and derives a modified loss function to train the critic for each agent based on its own reward. The proposed strategy is tested on datasets with a large agent population size to decide whether drivers should stay in the current zone or move to another zone to look for passengers for total profit maximization.

5.2. Order Dispatching

On the premise of ensuring available vehicles in various areas by vehicle re-positioning, the dispatching strategies to meet the large volume of orders in real-time are emphasized in a large number of studies. Traditional rule-based solutions for order dispatching require sophisticated hand-crafted parameter design but are only effective on simplified problem settings (Li et al., 2019), which motivates the utilization of Reinforcement Learning.

Oda and Joe-Wong (2018) examines the framework of DQN with the dispatch center as the agent to minimize the passenger waiting time and idle cruising time and reduce the number of requests that are not responded to. However, all idle vehicles need to sequentially decide their destinations which will increase computation time and decrease the dispatching efficiency. Thus, the following studies to be discussed consider the agent as the driver/vehicle to construct a multi-agent-based RL framework for order dispatching.

Multi-agent RL strategies for order dispatching are also examined with either cooperative or independent agents. Zhou et al. (2019a) illustrates that explicit cooperation among various drivers is helpless for order dispatching since each driver serves different orders with different starting times, duration, and destination grids. Thus, each driver/vehicle is regarded as an agent working independently in this proposed method to explore the environmental information of the current locations, including the number of idle vehicles, valid orders, and destinations. To maximize the accumulated driver income (ADI) and order response rate (ORR), Double-DQN is extended with Kullback-Leibler (KL) divergence optimization to select optimal orders for drivers. More studies (Xu et al., 2018; Li et al., 2019; He and Shin, 2019) held a different opinion with Zhou et al. (2019a), which demonstrate the necessity of coordination among drivers for order dispatching. In detail, Li et al. (2019) clarifies that active agents sharing orders in the same/nearby areas might select the same order according to their own policy, which may cause conflicts. Thus, different methods have been proposed to solve such an issue based on the RL framework. Specifically, Mean Field Reinforcement Learning (Yang et al., 2018) is adopted to evaluate the average response among agents for agents interactions where the average response is derived from the number of drivers arriving at the same neighborhood and available orders. He and Shin (2019) proposes a capsule-based Double-DQN for coordination policy learning where the capsule means a structured group of neurons (Sabour et al., 2017). The capsule construction helps the agent to analyze spatial (e.g., geographical distributions of demands and supplies) and temporal (e.g., weather conditions over time) relations and further learn the final policy. In addition, Xu et al. (2018) formulates the action-value function as a bipartite graph matching problem (i.e., the edge between one driver and one order is set as the action-value function). The Kuhn-Munkres (KM) algorithm (Munkres, 1957) is employed for optimization to ensure that each order is assigned to at most one driver and avoid conflicts.

The mass deployment of MOD systems shows great success and high profits in megalopolis, which motivates the popularization of MOD systems in tier-three cities, which lack data for optimization and management. Therefore, on the ride order dispatching problem, Wang et al. (2018) and Tang et al. (2019) propose transfer learning methods to enable knowledge transfer from source cities with sufficient historical data to target cities with limited historical data. Since travel patterns of different cities often share common spatial and temporal characteristics, reusing previously trained DQN models learned from source cities to determine the optimal policies for target cities can be flexible and useful. Three transfer learning methods are tested in these two studies, i.e., fine-tune (Hinton and Salakhutdinov, 2006), progressive network (Rusu et al., 2016), and correlated-feature progressive transfer (Wang et al., 2018).

The aforementioned studies dealing with order dispatching, vehicle re-positioning, and pricing independently may ignore the high correlations between them (Jin et al., 2019). Thus, Holler et al. (2019) and Jin et al. (2019) explore these two tasks (order dispatching and vehicle re-positioning) simultaneously with different RL frameworks and agents, where actions of agents include vehicle re-positioning without an order and orders serving. Chen et al. (2019) studies the pricing strategy and order dispatching jointly since the user decides whether to submit the order request after knowing the estimated price of the input trip (i.e., origin and destination) given by the MOD system. In detail, Holler et al. (2019) aims to maximize the revenue of each driver independently from driver-perspective and maximize the combined revenue across all drivers from system-perspective by using different reward specifications and optimization algorithms (i.e., DQN and PPO). The optimization results show that the driver-perspective system is more competitive than the system-perspective approach. It is noteworthy that most multi-agent-based RL methods designed for MOD systems management regard each driver/vehicle as an agent, which results in high computational costs due to a large number of agents. Based on the framework of Hierarchical RL, Jin et al. (2019) chooses the region as an agent where large districts are manager agents while small grids are worker agents to model the ride-hailing system. The goal of the manager agent is to maximize ADI and ORR based on observations and peer messages (i.e., features extracted from other manager agents). The worker agents generate actions (i.e., pick up orders or re-position) following the objective developed by its manager and own observations. The action value of order dispatching depends on environmental states (e.g., locations of drivers and passengers) and pricing strategies. Thus, the total expected reward of the pricing strategy is composed of expected driver income before order completion and actual driver income, which means the optimal pricing strategy also relies on order dispatching.

More recently, Manchella et al. (2021) presents a novel and valuable direction for joint goods delivery and ride-sharing service with deep RL methods. Using the status of available vehicles and pick-up requests, the proposed model adopts Double-DQN to find optimal dispatching policies for passengers pooling and goods delivery. The ride-sharing data collected from New York City taxicab and customer check-in traffic data from Google Maps give the opportunity for this work to verify that jointly serving passengers and goods can be cost-efficient and environmentally friendly.

5.3. Vehicle Routing

In ride-sharing systems, multiple orders and various passengers with similar itineraries can be handled simultaneously, which means that the policies for vehicle routing after order dispatching should be addressed and studied. The methods with computational complexity issues are hard to be applied in time-sensitive vehicle routing applications. RL has already shown strong capabilities in vehicle routing/navigation. Also, the training process of RL-based strategies can be conducted offline so that the route generation process can be handled handy and fast (James et al., 2019) in large transportation networks. Therefore, RL becomes an essential tool for vehicle routing in ride-sharing service systems.

RL strategies for vehicle routing in MOD systems include both single-agent algorithms (James et al., 2019) and multi-agent algorithms (Al-Abbasi et al., 2019; Silva et al., 2019; Zhang et al., 2020b). Specifically, the dispatch center is regarded as the agent in James et al. (2019) based on the formulation of green logistic systems (James and Lam, 2017). The Asynchronous Advantage Actor-Critic (A3C) method is adopted to train the route construction policy to serve more orders while minimizing the driving distances of all vehicles. To further explicitly study the cooperation or competition among vehicles or customers, Zhang et al. (2020b) regards each vehicle as an agent and designs a multi-agent attention RL-based model. The model consists of an encoder-decoder structure where the encoder module analyzes the relations among customers while the decoder module decides the choice of the next visited customer via reinforcing gradient estimator optimization. The optimization of vehicle routing independently neglects the correlations between order dispatching and vehicle routing, which motivates Al-Abbasi et al. (2019) to focus on providing policies for two tasks simultaneously via Double-DQN. Each vehicle works as an agent to decide whether to serve existing or new users after observing and analyzing the predicted future demand and the time cost before vehicles become available. If a new user is chosen or the vehicle is empty, the agent determines the zone to arrive. This study shows the superiority of ride-sharing in reducing traffic congestion through experiments on the real-world dataset from New York City.

Silva et al. (2019) determines a set of routes to make each customer can be served by one vehicle based on a single depot with Q-learning. In order to minimize the number of vehicles and reduce travel distances, the action is set to decide the locations and order of passengers to be served by acknowledging the information of all vehicles and customers.

6. Assistant and Autonomous Driving

Ensuring safety is the most critical objective in transportation systems for both human-piloted driving and autonomous driving. Driver-assistance systems (DASs) and autonomous vehicles (AVs) are expected to enhance driving safety and also improve traffic efficiency (Pan et al., 2021). In this section, a widely studied DAS technology, adaptive cruise control (ACC), with the strategies of Reinforcement Learning, is introduced first. Then, two types of training methods for decision-making modeling based on RL (i.e., car-following modeling to decide the velocity/acceleration and lane-changing modeling for steering control) are presented. A list of studies using RL for assistant/autonomous driving is provided in Table 5.

6.1. Adaptive Cruise Control

The technologies of driver-assistance systems have been embedded into vehicles to improve the driving experience and reduce traffic accidents. Adaptive cruise control (ACC), as an essential function of the system, has the ability to adjust the speed and acceleration of the current vehicle and further maintain a safe distance from the vehicle in front of it. To reduce reliance on prior knowledge of disturbance measurements (Li et al., 2017a), Reinforcement Learning becomes a valuable tool for ACC.

Adaptive cruise control with RL has been examined for both private vehicles and buses. As for the private vehicle, the speed and acceleration of the current vehicle and the distance from the front vehicle are collected as the state for adaptive cruise control policy optimization (Desjardins and Chaib-Draa, 2011; Li et al., 2017a; Li and Görges, 2019) with various reward functions and RL frameworks. Specifically, Desjardins and Chaib-Draa (2011) takes advantage of DDPG to determine the action (e.g., braking, accelerating). Li et al. (2017a) utilizes Q-learning to select the specific values of permissive accelerations, which can be more feasible in practice. Li and Görges (2019) investigates driving safety and fuel consumption simultaneously by optimizing the velocity and the online gear shift jointly. The utilized deep Actor-Critic framework consists of two actor networks and a critic network. Two actor networks are used to generate the traction force for velocity tracking and provide the gear position for fuel economy, respectively. And the critic network evaluates the control performance for these two purposes.

The investigation of the bus adaptive cruise control with RL has received less attention. Gao et al. (2019) proposes a cooperative ACC algorithm with a central controller for a fleet of autonomous buses on the exclusive bus lane (XBL). The policy iteration RL method is employed to approximate the value of the control gain introduced in the linear optimal control theory (Lewis et al., 2012). The experimental results show that the proposed method is able to increase the traffic throughput and save the travel time of buses.

More recently, Nascimento et al. (2021) reports that safe driving can be affected by the driver's comfort and feel, which can be adaptable for all types of vehicles. To investigate the interplay between the perceived sounds of a vehicle and the driver's attention/enjoyment, a psychoacoustic (PA) metric (Pedersen and Zacharov, 2008) is used as the reward function to measure the driver's feeling where lower PA values mean more comfort. The agent analyzes environmental sounds (e.g., pedestrians and traffic) and noises (e.g., sounds of bells and beeps) to decide the states of the window (no change, open, close), radio (no change, on, off), and speed (no change, accelerate, decelerate) with the optimization via Double-DQN. The proposed method has the ability to change the state of the vehicle to maintain the driver's concentration for driving safety.

6.2. Velocity and Acceleration Control

Velocity/acceleration control of the autonomous vehicle has the promise of improving traffic safety and increasing road capacity (Zhu et al., 2020), which has been studied in numerous studies with Reinforcement Learning.

Table 5
Summary of RL Applications in Assistant and Autonomous Driving

Reference	Application	Framework	Agent	Scenario/ Data	Simulator
Desjardins and Chaib-Draa (2011)	adaptive cruise control	DDPG	single-agent, a vehicle	synthetic network	personal simulator
Li et al. (2017a)	adaptive cruise control	Q-learning	single-agent, a vehicle	synthetic network	personal simulator
Li and Görges (2019)	adaptive cruise control	Deep Actor-Critic	single-agent, a vehicle	synthetic network	personal simulator
Gao et al. (2019)	adaptive cruise control for buses	Policy Iteration	single-agent, the center	Lincoln Tunnel Corridor	Paramics
Nascimento et al. (2021)	drivers' comfort modeling	Double-DQN	single-agent, a vehicle	synthetic network	GTA V simulator [1]
Zhu et al. (2018)	acceleration control	DDPG	single-agent, a vehicle	synthetic network	personal simulator
Zhou et al. (2019b)	acceleration control	DDPG	single-agent, the center	synthetic network	personal simulator
Zhu et al. (2020)	velocity control for electric vehicle	DDPG	two agents, following and lead vehicle	NGSIM dataset [2]	Next Generation Simulation [2]
Wegener et al. (2021)	acceleration control	Twin-delayed DDPG	single-agent, a vehicle	NGSIM dataset	Intelligent Driver Model
Liu et al. (2021)	lane keeping	DDPG	single-agent, a vehicle	real and synthetic scenarios	simulator from OpenAI Gym
Cao et al. (2020)	acceleration control and lane changing for highway existing	Monte Carlo Tree Search	single-agent, a vehicle	synthetic network	personal simulator
Ye et al. (2019)	acceleration control and lane changing	DDPG	single-agent, a vehicle	synthetic network	VISSIM
Guo et al. (2021)	acceleration control and lane changing	DDPG	single-agent, a vehicle	synthetic network	SUMO
Sathyan et al. (2021)	acceleration control and lane changing	DQN	multi-agent, a vehicle	synthetic network	SUMO
Pan et al. (2021)	ramp metering, lane changing, speed limit control	Cross-Entropy-Method	single-agent, a vehicle	synthetic network	personal simulator
Wachi (2019)	failure scenario finding	DDPG	multi-agent, a vehicle	synthetic network	Microsoft AirSim, (Shah et al., 2018)

[1] https://github.com/aitorzip/DeepGTAV
[2] https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Zhu et al. (2018) introduces an autonomous driving model based on DDPG to reproduce behaviors and trajectories of drivers. To determine the acceleration of the vehicle, the agent sets the reward function as minimizing the disparity of spacing and velocity between the simulated and observed data. Note that solely imitating human driving behaviors for autonomous vehicles may not reduce traffic accidents or increase road capacity due to the hardly optimal operation of human drivers (Zhu et al., 2020). Thus, the following studies (Zhou et al., 2019b; Zhu et al., 2020; Wegener et al., 2021) directly optimize autonomous driving from interactions with the simulated environment (i.e., surrounding vehicles information, own driving information, and road networks) by adopting various deep RL strategies under different scenarios. As for the framework of RL, DDPG is adopted in Zhou et al. (2019b); Zhu et al. (2020) while Twin-delayed Deep Deterministic Policy Gradient (TD3) (Fujimoto et al., 2018) is used by Wegener et al. (2021). As for the application scenarios, Zhou et al. (2019b) and Wegener et al. (2021) focus on obtaining appropriate driving acceleration under different levels of traffic and lengths of the signal cycle at intersections. Zhu et al. (2020) examines velocity control of autonomous driving under different road incidents/events, which improves safety, efficiency, and comfortableness, as shown by their experimental results.

6.3. Steering Control and Lane Changing

Keeping the vehicle within the lane and driving stably are essential for the safety of autonomous driving (Liu et al., 2021). Liu et al. (2021) collects the distances from the vehicle to the road lane borders from the GPS information as the state to decide the vehicle's steering angle via the framework of DDPG. To accommodate the real-world scenario with information noise, a noise compensation approach is used. The independent optimization of steering control or lane changing can be less practical since the change in position often results in the change in velocity or acceleration. Thus, many studies determine longitudinal and lateral positions simultaneously to achieve safer and more efficient autonomous driving.

Initial works only depend on one optimizing strategy for two tasks (Ye et al., 2019; Cao et al., 2020; Sathyan et al., 2021). In detail, in order to increase the success rate of exiting from highways in heavy dynamic traffic, Cao et al. (2020); Sathyan et al. (2021) optimize longitudinal acceleration and the policy of lane changing by Monte Carlo Tree Search (Browne et al., 2012) and DQN, respectively, where the distance to the exit ramp and the surrounding vehicles' positions and speeds are regarded as the state. Ye et al. (2019) proposes a more general strategy to decide the longitudinal and lateral position of the vehicle jointly under different driving environments based on DDPG with the driving information of surrounding vehicles. The reward form is calculated by its distance from the preceding vehicle, its speed, and the speed difference to the preceding vehicle. The collision, uncomfortableness, and inefficient driving performances are also penalized in the reward. Guo et al. (2021) finds the optimal policies for the continuous longitudinal acceleration/deceleration and discrete lane changing via DDPG and DQN, respectively. The two optimizing strategies are able to interact with each other and reduce the error probability, which is more robust in unusual driving conditions (e.g., abrupt deceleration of the front vehicle).

Furthermore, an integrated model is proposed to deal with more comprehensive tasks, i.e., ramp metering, variable speed limit, and lane changing control for both connected autonomous vehicles and regular human-piloted vehicles to minimize the total travel cost in Pan et al. (2021). The proposed model is optimized by the gradient-free Cross-Entropy-Method-based algorithm (Szita and Lörincz, 2006).

In addition, a novel way to deal with the safety of autonomous driving is introduced (Wachi, 2019), i.e., identifying failure scenarios of the vehicle. The environment consists of two types of vehicles, the player and multiple non-player characters (NPCs). And the aim is to train NPCs to make the player cause an accident or arrive at the destination late. When the player fails, NPCs get the adversarial reward based on their own contributions to the failure. Multi-agent DDPG algorithm (Lowe et al., 2017) is employed to train the agents to find the optimum driving directions and velocity. Their strategy provides a novel and effective direction to avoid catastrophic accidents for autonomous driving.

Table 6
Summary of RL Applications in Routing

Reference	Application	Framework	Agent	Scenario/Data	Simulator
Cao et al. (2017)	path recommendation	DQN	single-agent, the driver	networks of Munich, Singapore, Beijing	personal simulator
Ramos et al. (2018)	routing for travel time minimization	Q-learning	multi-agent, the driver	synthetic data	personal simulator
Boutilier et al. (2018)	shortest path routing	DQN	single-agent, the driver	network in San Francisco Bay Area	personal simulator
Chandak et al. (2020)	shortest path routing	Policy Gradient algorithm	single-agent, the driver	network in San Francisco Bay Area	personal simulator
Mao and Shen (2018)	routing for travel time minimization	Neural fitted Q-iteration, Q-learning	single-agent, the driver	Sioux Falls network	personal simulator
Zhang and Masoud (2021)	GPS correctness	A3C	single-agent, the controller	GPS trip recorder in Southeast Michigan	personal simulator
An et al. (2020)	routing for travel time minimization	DQN	single-agent, the controller	synthetic data	personal simulator
Zhang et al. (2019b)	parking	DDPG	single-agent, the controller	synthetic data	personal simulator
Wang et al. (2021b)	parking	Monte-Carlo	single-agent, the controller	synthetic data	personal simulator

7. Routing

RL-based vehicle routing in taxi, ride-sourcing, and ride-sharing systems have been reviewed in Section 5. This section discusses RL-based routing in a more general context, where routing plays an important role in both human-driving and autonomous driving vehicles. It should be noted that the accuracy of Global Positioning System (GPS) localization is critical in vehicle navigation/routing applications, which might be affected by environmental factors (e.g., weather and occlusion of buildings). Raw GPS observations (i.e., longitude and latitude coordinates) are corrected in Zhang and Masoud (2021) by the algorithm of A3C, where the state is the observation history trajectory consisting of the last reported position and the most recent predicted positions within a certain period. Many previous studies on routing problems are based on parametric models with strong behavior assumptions (Mao and Shen, 2018). Tail-based research (Lim et al., 2013) for routing often suffers from the issue of low accuracy and high computational cost. Instead, given its capability for optimal policy discovery without expert knowledge and its scalability for adapting the proposed methods to large-scale real-world networks, RL-based models have been used to find the shortest path and minimize total travel time. This section mainly introduces routing problems from two aspects, i.e., the stochastic shortest path problem and real-time routing. The introduced RL-based works for routing are summarized in Table 6.

The stochastic shortest path (SSP) problem with RL is first studied in Cao et al. (2017) by adopting Q-learning as the framework and designing a deep-based approximator to represent the value function for adaptation to large road networks. In practice, some travel paths are not always reachable due to road construction or other reasons, which motivates the exploration of the unavailability of actions by introducing stochastic action sets (SAS) (Boutilier et al., 2018). DQN is adopted as the framework to illustrate the effects on the shortest path sought problem with the consideration of the probability of the shortest path availability. The results indicate that the optimal policy with SAS has the ability to yield an expected travel time between the origin and destination within a target small range. Following studies (Boutilier et al., 2018; Chandak et al., 2020) further examine each node as the origin and learns the shortest path from each node. The proposed framework generalizes the Policy Gradient algorithm to estimate the optimal policy in a large-scale network.

RL methods for the SSP problem build the foundation for the real-time routing strategy, which needs to minimize the expected total travel time by accounting for real-time traffic conditions. Therefore, continuous variables describing real-time traffic congestion are used in many studies (Ramos et al., 2018; Mao and Shen, 2018; An et al., 2020) to look for the path that minimizes travel time or travel delay based on different developed RL strategies. The adaptation of Q-learning is combined with the regret-minimising method in Ramos et al. (2018) to minimize travel time for routing. The Neural Fitted Q Iteration (FQI) (Ernst et al., 2005) is adopted in Mao and Shen (2018) to accommodate the large state space (i.e., the constantly changing instantaneous travel cost) and produce a more refined representation of the Q-function for further routing policy optimization. An et al. (2020) utilizes DQN with the help of the Dijkstra algorithm and k-shortest path algorithm to determine the platoon size on the monitor link where the platoon strategy is used to avoid conflict points in platoons for routing assistance.

Moreover, routing for parking issues has been discussed in Zhang et al. (2019b); Wang et al. (2021b). Specifically, Zhang et al. (2019b) adopts DDPG for autonomous parking (i.e., determine the steering wheel angle) with the coordinates of the four corner points in the vehicle. Wang et al. (2021b) proposes a Monte-Carlo-based optimization model on parking spot selections, which becomes a crucial problem in mega-cities for automated multistory parking facilities. In order to reduce customers' waiting time, the agent is in charge of choosing the parking level for each vehicle on the elevator by analyzing the status of available parking spots and the current time.

8. Public Transportation and Bike-sharing System

The public transportation system (e.g., buses and trains) and bike-sharing system serve a large number of passengers and play a vital role (Li et al., 2021a) in metropolitan areas for environmental protection. RL-based strategies have been examined for public transit and bike-sharing systems scheduling and management to improve efficiency and profitability, which are reviewed in this section. A summary of the papers to be discussed is provided in Table 7.

8.1. Bus Holding

Bus holding, a strategy that delays buses at control points (Dai et al., 2019), has received substantial attention for many decades in order to reduce the probability of bus delay, decrease the waiting/travel time of passengers, and thus improve the efficiency of the bus system (Berrebi et al., 2018). A large number of strategies mainly consider local information with a pre-specified headway/schedule. However, the global coordination of the whole bus fleet and the long-term effect are often overlooked (Wang and Sun, 2020), which can be potentially addressed by RL-based methods.

Owing to the mutual influence among buses, existing studies (Chen et al., 2016; Alesiani and Gkiotsalitis, 2018; Menda et al., 2018; Wang and Sun, 2020) adopt different multi-agent RL frameworks by regarding each bus as an agent to analyze the input state (e.g., treating departure time, arrival time, and target headway time of the bus) and determine bus holding duration with different granularity. Specifically, 30 seconds is set as the minimum unit of holding time in Alesiani and Gkiotsalitis (2018) with the optimization by Double-DQN. Since the bus holding time less than 30 seconds is not practical considering constraints from real-world driving conditions, the holding time is chosen as some multiple of the holding time unit (e.g., 30 seconds) in Chen et al. (2016) optimizing by Q-learning and Menda et al. (2018) optimizing by PS-TRPO (Gupta et al., 2017). Though these methods adopt multi-agent frameworks to deal with holding time for multiple buses simultaneously, less attention has been paid to agents' cooperation. More recently, Wang and Sun (2020) proposes a global joint action tracker embedding into the PPO framework to incorporate global coordination for dynamic bus holding control. The action tracker network is used to adopt the global information of buses and passengers to further track the policies of each agent (i.e., a bus). Thus, the state evaluation of each agent's policy is based on the local environment and other agents' decisions.

Table 7
Summary of RL Applications in Public Transportation and Bike-sharing System

Reference	Application	Framework	Agent	Data	Simulator
Alesiani and Gkiotsalitis (2018)	bus holding	Double-DQN	multi-agent, a bus	a main bus line in Singapore	personal simulator
Chen et al. (2016)	bus holding	Q-learning	multi-agent, a bus	synthetic data	personal simulator
Menda et al. (2018)	bus holding	PS-TRPO	multi-agent, a bus	synthetic data	personal simulator
Wang and Sun (2020)	bus holding	deep PPO	multi-agent, a bus	synthetic data	personal simulator
Yin et al. (2014)	acceleration control for the subway	Q-learning	single-agent, a subway	real data from Beijing Subway	personal simulator
Yang et al. (2021)	voltage control for urban railway	DQN	single-agent, the center	real data from Beijing Subway	personal simulator
Šemrov et al. (2016)	train scheduling	Q-learning	single-agent, the center	railway network in Slovenia	personal simulator
Khadilkar (2018)	train scheduling	Q-learning	single-agent, the center	railway lines from Indian	personal simulator
Ying et al. (2020)	subway scheduling	DDPG	single-agent, the center	London Underground	personal simulator
Jiang et al. (2018)	inflow control for urban rail transit	Q-learning	single-agent, the center	metro line in Shanghai	personal simulator
Wei et al. (2020)	next metro line design	Deep Actor-Critic	single-agent, the center	the current metro network in Xi'an	personal simulator
Li et al. (2018)	bike re-position for bike-sharing system	DQN	multi-agent, a trike	Citi Bike data from New York	personal simulator
Pan et al. (2019)	price management for bike-sharing system	DDPG, Hierarchical RL	multi-agent, a user	Mobike dataset from Shanghai	Mobike's original system

8.2. Urban Rail Transit System Management

Adopting the mechanism of Reinforcement Learning, multiple research topics have been investigated for the operation of the urban rail transit system (e.g., train and subway), such as energy management, vehicle re-scheduling, passenger flow control, and network expansion which will be introduced in this subsection.

Energy management: A few studies aim to use RL method to minimize the energy consumption of subway operation where two optimizing types are proposed, i.e., managing one subway vehicle independently and managing the whole subway system. In detail, Yin et al. (2014) defines the current vehicle position, the speed, and the reserved trip time as the state and each subway vehicle as an agent to decide the variation of acceleration via Q-learning. In order to cooperate with other subways to acknowledge the time-vary traffic, Yang et al. (2021) uses the super-capacitor energy management system (SCESS) as the central agent for energy-saving and voltage stabilization of the whole subway system. The states of the subways nearing the SCESS and the rectifier current/voltage of the substation where the SCESS is installed are accounted for the state in the implementation of RL. And the agent decides on the combination of charging and discharging voltage threshold to increase the energy-saving rate and voltage stabilization rate in each time step.

Scheduling: Scheduling is one of the core issues for urban rail transit systems, e.g., in order to reduce the travel/waiting time and the operating cost (Zhao et al., 2021). Train scheduling for both the single-track railway (Šemrov et al., 2016) and multi-track railway (Khadilkar, 2018) are examined. The information in relation to the locations of trains, the infrastructure availability of block sections, and the time is considered in Šemrov et al. (2016) for single-track railway scheduling. Q-learning is used to decide the actions for each signaling element, i.e., setting it to red (stop) or green (go) color, indicating which trains can move on to the next section, which helps reduce the total delay effectively. However, the study dealing with the single-track railway cannot be directly adapted to multi-track railway systems (e.g., the trains operating on multiple tracks can be merged into one track which may cause disruption). Train scheduling on multi-track is taken into consideration by the study of Khadilkar (2018), where directions of trains' motion are analyzed for further decision-making with Q-learning. Different to train scheduling, urban subway scheduling has to take the number of passengers into account for decision-making (Ying et al., 2020). The optimizing framework based on DDPG shows very satisfactory performance in terms of reducing passenger waiting times and saving subway operating costs.

Passenger flow control: To decrease the waiting time of passengers and reduce accidents caused by crowds in railway stations, the control of passenger inflow for railway systems has been investigated in Jiang et al. (2018). The environmental state includes information of real-time passenger demand, the arrival/departure time, the available capacity of trains, and the platform capacity of stations. Q-learning is adopted to set the rate of inflow volume for each station. The experimental results show that inflow control with RL can reduce the number of passengers being stranded and relieve passenger congestion at certain stations.

Network expansion: The design or the expansion of a railway transit network is another primary concern in public rail/transit systems (Laporte et al., 2010). Most existing strategies dealing with network expansion are often based on conventional mathematical programming approaches, which are heavily dependent on expert guidance and behavior assumptions (Wei et al., 2020). Instead of the usage of domain knowledge and behavior assumptions, the Actor-Critic framework with single-agent is adopted in Wei et al. (2020) to select the locations of expanded stations in the city metro network. Specifically, the actor network is an Encoder-Decoder Neural Network coupling with an attention layer to parameterize the station selection policy for metro line expansion, while the critic network consists of three convolutional layers and two fully connected layers to estimate the expected cumulative reward of the next metro line.

8.3. Bike-sharing System

Bike-sharing systems, including dock and dock-less systems, are widely deployed in urban and rural areas to ease the first/last-mile problems and reduce the usage of private vehicles. Li et al. (2018) and Pan et al. (2019) aim to balance the supply and demand of these two systems, respectively. In order to minimize the customer loss of the system with dock, Li et al. (2018) proposes a multi-agent DQN-based bike re-positioning method. Each trike (i.e., the tool for moving bikes) is regarded as the agent that chooses the location of the station and the number of picking up or unloading bikes after observing the system status (i.e., bike and dock availability at each station), its own status (i.e., the available location for bikes), and the status of other trikes. Pan et al. (2019) focuses on pricing management to incentive users for the dock-less bike-sharing system. Building upon DDPG and Hierarchical RL, the proposed pricing algorithm suggests the user return the bike to neighboring regions by offering a price incentive under a default budget.

9. Electric Vehicle: Energy Management, Charging, and Ride Service

To mitigate the crisis of resource scarcity and climate change, electrification has been the trend of the automotive industry to achieve the merits of high performance and long-term economy (Wu et al., 2020a). Reinforcement Learning methods have been adopted for electric vehicle (EV) control and management in recent years, especially for ground electric vehicles. This section mainly introduces the RL applications on two major ground vehicles, hybrid-electric vehicles (HEVs) and pure-electric vehicles (PEVs). The mentioned works in this study are summarized in Table 8.

9.1. Hybrid-Electric Vehicle

A hybrid-electric vehicle usually combines a conventional powertrain (e.g., gasoline) with an electric engine. Most existing studies dealing with energy management of HEVs follow pre-defined rules, which heavily rely on the accurate prediction of future traffic conditions and are not straightforward for applications under time-sensitive driving conditions (Qi et al., 2019). RL strategies have been effective tools to avoid the need for precise forecasts.

Table 8
Summary of RL Applications in Electric Vehicle

Reference	Application	Framework	Agent	Data	Simulator
Liu et al. (2015a)	fuel and electricity sources control	Q-learning	single-agent, a vehicle	synthetic data	MotoTune [1]
Qi et al. (2016)	fuel and electricity sources control	Q-learning	single-agent, a vehicle	inductive loops detector data archived in the California Freeway PEMS [2]	Motor Vehicle Emission Simulator [3]
Liu et al. (2017)	fuel and electricity sources control	Q-learning	single-agent, a vehicle	synthetic data	personal simulator
Qi et al. (2019)	fuel and electricity sources control	DQN Dueling-DQN	single-agent, a vehicle	inductive loops detector data archived in the California Freeway PEMS [2]	personal simulator
Wu et al. (2019)	fuel and electricity sources control	DDPG	single-agent, a vehicle	synthetic data	Paramics
Lian et al. (2020)	fuel and electricity sources control	DDPG	single-agent, a vehicle	synthetic data	personal simulator
Wan et al. (2018)	EV charging/ discharging scheduling	DQN	single-agent, a vehicle	real scenario from the California ISO	personal simulator
Zhang et al. (2021a)	EV charging/ discharging scheduling	DQN	single-agent, a vehicle	real data from EV charging stations data in Beijing	personal simulator
Luo et al. (2020)	EV re-positioning	PPO	multi-agent, a hexagonal grid	real EV sharing data in Shanghai	personal simulator
Shi et al. (2019)	EV dispatching and charging management	DQN	multi-agent, a vehicle	synthetic data	personal simulator
Tang et al. (2020)	EV taxi-customer assignments, vehicle dispatching and charging	Deep RL	single-agent, a central controller	real data from Tongzhou and Beijing	personal simulator
Zhang et al. (2020c)	EV route planning and energy management	Actor-Critic, Q-learning	single-agent, the controller	synthetic data	ADVISOR [4]
Lin et al. (2021)	vehicle routing for Electric Vehicles	REINFORCE	single-agent, the controller	synthetic data	personal simulator

[1] http://mcs.woodward.com/support/wiki/index.php?title=MotoTune
[2] http://pems.dot.ca.gov
[3] https://www.epa.gov/moves
[4] http://bigladdersoftware.com/advisor/docs/advisor_doc.html

The studies start to regard the energy management center as the agent for engine power control via Q-learning in Liu et al. (2015a); Qi et al. (2016); Liu et al. (2017) with different state settings. In detail, Liu et al. (2015a) explores the knowledge of environmental features, the battery state-of-charge (SOC), and the rotational speed of the generator (i.e., engine speed) to determine fuel consumption. More related characteristics are analyzed in Qi et al. (2016), i.e., the vehicle velocity, road grade, percentage of remaining time to destination, SOC, and available charging gain of the selected charging station. The internal combustion engine (ICE) power supply level (discrete form) obtained from the optimization is chosen to further control the proportions of electricity and fuel to use. The predicted future velocity profile and the information of SOC are utilized in Liu et al. (2017) as the state to select the throttle engine power and further determine the power distribution of the electrical energy source and conventional powertrain source. The velocity profile is obtained by two novel velocity predictors (i.e., Nearest Neighbor Velocity Predictor and Fuzzy Encoding Velocity Predictor).

A number of deep RL studies have shown their capability to handle non-linear and complicated relations among vehicles and the traffic environment for traffic control, which motivates the utilization of deep learning in energy management. Complex and powerful deep RL methods are proposed to control electricity and conventional powertrain energy split for HEVS with different reward functions and state settings (Qi et al., 2019; Wu et al., 2019; Lian et al., 2020). In detail, Qi et al. (2019) uses DQN and Dueling-DQN to select an optimal fuel/electricity split's level (i.e., 24 power level outputs are set for the engine) with the information regarding the power demand at the wheel, the battery pack's state-of-charge, and the distance to the destination to reduce fuel consumption. This study optimizes the agents based on a single driving cycle that might not be able to deal with different driving cycles (DCs) or the entire driving profile of a vehicle (Wu et al., 2019). Therefore, Wu et al. (2019) adopts the framework of DDPG to model the energy split management for multiple driving cycles. Given the control variables (e.g., rational speed of engine/motor) as the current state of the environment, the actor network represented by the structured control net (SCN) (Srouji et al., 2018) produces an action while the critic network consisting of several fully connected layers estimates the action-value function. Moreover, considering that human expertise can provide optimal training samples or preferences for the learning agent to guide exploration in the training process, Lian et al. (2020) proposes a rule-interposing DDPG model to deal with the time-consuming problem caused by deep RL strategies. The added expert knowledge includes the optimal brake specific fuel consumption curve of the HEV engine and the battery characteristics, which helps set control variables of RL models. The aim of the controller is to optimize the engine power increment or decrement (e.g., remain unchanged, increase one kilowatt, decrease one kilowatt).

Different from the aforementioned studies focusing on energy management and splitting independently, Lin et al. (2021) adopts the Actor-Critic framework and Q-learning for route planning with power management of plug-in HEVs to minimize energy consumption. The inner loop is in charge of managing power by controlling the desired output torque from the engine, the gear shift command, and the direction by analyzing the state (i.e., vehicle status and geographic information). Meanwhile, the outer loop decides the changes in road slope and vehicle speed, which can affect energy utilization. The overall reward is designed to minimize fuel consumption and battery recuperation instead of only considering the shortest distance between the origin and the destination.

9.2. Pure-Electric Vehicle

The usage of pure-electric vehicles is rapidly growing, while the driving range and insufficient charging stations of EVs are two adverse factors on the widespread adoption of pure-electric vehicles (He et al., 2018). In order to solve such issues, recently, DQN-based frameworks are designed for EV charging/discharging scheduling subject to different objectives (Wan et al., 2018; Zhang et al., 2021a). Wan et al. (2018) aims to improve user benefit by designing a representation network to extract discriminative features from the battery state-of-charge (SOC) and the future price trends predicted by Long Short-Term Memory (LSTM). The Q-network is utilized to approximate the optimal action-value function and then make the decision for the amount of energy that the EV battery will be charged or discharged. Zhang et al. (2021a) aims to minimize the total charging time of EVs and reduce the distance between the origin and charging stations. The

EVs charging schedule system analyzes the features from the available charging piles and the EVs electricity consumption (predicted by distance traveled with linear regression) to obtain Q-value for selecting a charging station for the vehicle.

Pure-electric vehicles have also been introduced to provide ride-sourcing services with the fast improvement of battery technologies and the rapid growth of recharging facilities (Kim et al., 2015; Ke et al., 2019). As presented in Section 5, a number of RL-based methods have been put into use for dispatching and routing gasoline vehicles, which can also be adapted for ride-sourcing management of EVs. Different from conventional gasoline vehicles, EV re-position, dispatching, and routing often more explicitly take into account the recharging or electricity consumption issues of EVs.

Specifically, unbalanced/skewed distributions of EV fleets motivate Luo et al. (2020) to propose a multi-agent RL model for EV re-positioning in order to improve demand rate and net revenue. The designed actor-critic-based PPO model consists of two connected policy networks, one used for choosing the grid and another adopting the output from the first network for further selecting the station in the chosen grid with the agent (i.e., each hexagon grid of the urban area in concern). The proposed model can deal with the non-stationarity in action spaces caused by the station extension or closure by the regularization of the reward function.

Vehicle dispatching considering an electric vehicle fleet has also been studied (Shi et al., 2019; Tang et al., 2020; Lin et al., 2021) with different RL frameworks and optimizing aims. Shi et al. (2019) designs a DQN-based algorithm to dispatch the electric vehicle for ride-hailing services in terms of reducing EV operational costs and customer waiting time. The proposed framework consists of two components: the decentralized learning process to approximate the state-value function with the knowledge of vehicles and dispatching tasks; the centralized decision-making process to formulate and maximize the state-value function for EV fleets by a linear assignment problem and further to find the optimal dispatching policy. Tang et al. (2020) designs a two-step framework, advisor-student RL, to dispatch vehicles and arrange charging activities. In the advisor network, the control center assigns the status of vehicles (i.e., to be charged or to accept the order) to minimize the system cost (i.e., customer waiting cost, customer abandon penalty, vehicle travel cost, and vehicle charging cost) through the optimization by DQN. The student network decides the vehicle-customer pair and vehicle-charging-station pair via assignment problem optimization. Lin et al. (2021) focuses on reducing total distances of electric vehicles by solving routing problems (i.e., choosing the geographical coordinate of the next location) with the REINFORCE algorithm (Williams, 1992).

10. Future Directions and Conclusion

In the past decade, we have seen a growing number of studies that develop/adapt Reinforcement Learning methods for applications in the transportation sector. However, the development and utilization of advanced RL strategies for a more efficient and sustainable transportation system are still at an early stage. This section will discuss several aspects that deserve substantial further efforts in terms of developing RL methods for real-world transportation applications, i.e., scalability, practicality, transferability, and fairness.

Scalability:

- Existing RL-based studies for transportation applications are often capable of dealing with a single subject and/or one aspect of the system (e.g., speed limit control for a target part of the freeway (Zhu and Ukkusuri, 2014)). The demand for computing resources and computing time can be extremely high when adapting these methods to multiple-object large-scale environments, especially where there are complex interactions among objects or sub-systems within the system (e.g., a city often is served with thousands of intersections). Developing competent models with a cooperative and/or competitive multi-agent RL-based framework to deal with multi-object large-scale transportation systems is crucial. For instance, handling a single train in urban rail transit system management will be more feasible given the current development of RL methods, while optimizing the whole system with a large number of objects (or agents) will be much more challenging. Developing a scalable model with the ability to adopt and analyze large-scale spatial-temporal features and jointly optimize the actions of multi-object requires substantial novel efforts and innovations. For example, hierarchical RL can be a promising concept for handling such large-scale problems with a centralized manager for overall control and optimization and multiple decentralized workers for implementations at the local level.

Practicality:

- The design of the environment and reward function is critical for RL-based methods. Many methods are evaluated based on simulations with simulated observations and rewards. Only several works take advantage of real-world platforms for evaluation (e.g., Zhou et al. (2019a) uses the platform provided by Didi Chuxing for optimizing order dispatching). A certain (and unknown) gap between simulation and reality may exist. It is essential to train and evaluate the proposed methods based on real-world environments for policy optimization. For instance, order dispatching for MOD systems might be tested on real-world platforms such as Uber and Didi Chuxing so that the actual values of order response rate, driver income, and waiting/travel time can be obtained. Also, the utilization of a digital twin framework to mimic the real transportation system as a virtual system can be helpful in obtaining more realistic feedback. This often requires coordinated and cooperative efforts from academia, industry, and government.

- Existing studies are able to accommodate soft constraints effectively by introducing the penalty to reward functions. For instance, Tang et al. (2020) introduces a customer abandon penalty to reduce the possibility of order cancellation. The hard/rigid constraints of the environment are sometimes not straightforward to be incorporated, which should be investigated in future studies. This might require proper designs of environments and actions with limitations. For instance, the number of moving bikes in the bike-sharing system cannot exceed the capacity of the trike (the tool for moving bikes), which can be achieved by designing the range of the action vector.

- The evaluation of RL methods is sometimes based on ideal simulated environments (e.g., bus holding without considering the sluggish of passengers (Alesiani and Gkiotsalitis, 2018)). In practice, uncertainties, disruptions, and accidents often occur for road traffic, rail traffic, and air traffic. External factors which may influence the transportation system and network traffic should be analyzed or predicted (e.g., accurate weather forecasting can effectively help aircraft scheduling), and then incorporated for more capable RL tools.

- Some information such as travel demand, traffic flow, vehicle speed, trip distance, and trip time might be simulated or estimated for further decision-making with RL methods. For example, Citi Bike demand data from New York is collected in Li et al. (2018) for bike re-positioning. However, precise information in terms of some specific characteristics in the environment may not be readily available or hard to be obtained. For instance, some existing research for energy management of electric vehicles may require precise information regarding the drivers' behaviors, which might not be available at the time of decision-making. Therefore, some estimations or expectations might have to be assumed or further methods without such information request have to be developed (Qi et al., 2019; Wu et al., 2019).

- Some existing methods use discrete formulations for environmental features (e.g., the level of traffic congestion) and actions (e.g., slow down or speed up in adaptive cruise control (Desjardins and Chaib-Draa, 2011)), which achieve satisfactory performance based on private and public simulators. This is likely not universal and might not be sufficient in many real-world occasions. Inappropriate extensions of such methods to other applications might not be feasible or might result in low quality solutions. It is necessary to develop methods that are able to deal with the continuity and granularity of actions in transportation and optimize the choice of continuity and granularity since different scenarios require continuous or discrete actions with different (optimal) granularity. For example, the acceleration and steering control for autonomous driving requires extremely precise decisions since a slight adjustment in steering may cause a large change in the direction of a vehicle in the case of high-speed driving. On the contrary, it might be less meaningful to have a holding time for buses of ten seconds (while ten seconds might be too long for autonomous driving applications).

- The isolated design of different types of actions may also limit the practicality of RL to solve more complex transportation problems with substantial endogeneity or correlations among actions. Studies dealing with only one or two specific aspects of autonomous driving (e.g., lane changing, motion control, and collision avoidance) are still not ready for practical applications. More comprehensive consideration of multi-type actions simultaneously can be critical and essential in solving more complicated transportation problems in future research (e.g., to ensure safe, reliable, and efficient autonomous driving, the velocity, acceleration, angle change, route, and passengers' preference might have to be examined in an integrated manner).

Transferability:

- Studies targeting on existing road networks and public transit routes/stations have shown great success in numerous aspects, such as train scheduling (Khadilkar, 2018) and routing (Mao and Shen, 2018). Due to urban expansion, new transportation facilities have to be designed and arranged in existing or new regions, which receives less attention in the literature. The construction of new facilities requires sufficient expert knowledge due to the scarcity of historical data for policy optimization in RL. The utilization of transfer learning (Pan and Yang, 2009) and Meta-based RL (Finn and Levine, 2018) (i.e., the combination of Meta-Learning and Reinforcement Learning) are potentially effective tools for addressing new tasks or applications that lack sufficient training data. These strategies are able to transfer/adapt the trained RL-based model parameters/policies learned from the regions that already have related facilities to the new model for new regions.

Fairness:

- Existing studies aiming at improving the efficiency, profit, and safety of transportation systems by utilizing RL methods have made promising progress. However, the fairness issue of transportation systems has not been considered much, and is indeed often ignored in the development of RL methods. Different targets or entities (e.g., intersections or vehicles) may have to be fairly treated in the formulation of RL. To better address fairness issues in transportation, exploring the combination of survey data (stated preference) and other multi-source data is necessary. How to incorporate such a combination of data into RL method development is a direction that is worth further examination. Therefore, combinational weighted rewarding optimization problems with multiple objectives might have to be considered and addressed in transportation applications to achieve both efficiency and fairness. Effective combinational weighted rewards are not straightforward to be designed (e.g., the safety, efficiency, and comfort in autonomous driving are hard to be evaluated simultaneously), which might have to be solved by introducing other new algorithms or methodologies. For instance, Inverse Reinforcement Learning may be an effective solution to learn the reward function based on the agent's decisions and then find the optimal policy (e.g., Lanzaro et al. (2022) takes advantage of Inverse RL to recover the reward function of motorcyclists based on their actual trajectories for traffic conflicts modeling).

Reinforcement Learning and smart transportation are research topics that attracted substantial interest in recent years, where we see a large number of novel developments on strategies, techniques, and applications of RL to support smart transportation. It is also noted that applications of Reinforcement Learning in some sub-domains of transportation are limited, e.g., air traffic control and the aviation sector. For these application sub-domains, examining relevant and useful features is necessary.

In summary, this paper first uses the bibliometric analysis to identify the development of RL methods for transportation applications in recent years and then provides a review of the most relevant works covering a wide range of topics. This review provides readers with an understanding of RL-based method developments and applications in smart transportation and can serve as a reference point for researchers interested in interdisciplinary Reinforcement Learning research in transportation and computer science.

Acknowledgments

The authors would like to thank all anonymous referees for their thoughtful and constructive comments, which have helped to improve this paper substantially. Dr Liu would like to acknowledge the support from The Hong Kong Polytechnic University (P0039246, P0040900, P0041316), and the NSFC/RGC Joint Research Scheme (N_PolyU521/22).

References

Abdoos, M., Mozayani, N., and Bazzan, A. L. (2011). Traffic light control in non-stationary environments based on multi agent q-learning. In *14th International IEEE Conference on Intelligent Transportation Systems*, pages 1580–1585. IEEE.

Abdulhai, B. and Kattan, L. (2003). Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*, 30(6):981–991.

Abdulhai, B., Pringle, R., and Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285.

Al-Abbasi, A. O., Ghosh, A., and Aggarwal, V. (2019). Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4714–4727.

Alesiani, F. and Gkiotsalitis, K. (2018). Reinforcement learning-based bus holding for high-frequency services. In *2018 21st International Conference on Intelligent Transportation Systems*, pages 3162–3168. IEEE.

An, Y., Li, M., Lin, X., He, F., and Yang, H. (2020). Space-time routing in dedicated automated vehicle zones. *Transportation Research Part C: Emerging Technologies*, 120:102777.

Aradi, S. (2022). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):740–759.

Arel, I., Liu, C., Urbanik, T., and Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2):128–135.

Aslani, M., Mesgari, M. S., and Wiering, M. (2017). Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, 85:732–752.

Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., and Kautz, J. (2017). Reinforcement learning through asynchronous advantage actor-critic on a gpu. In *5th International Conference on Learning Representations*, pages 1–12.

Balaji, P., German, X., and Srinivasan, D. (2010). Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems*, 4(3):177–188.

Balakrishna, P., Ganesan, R., and Sherry, L. (2010). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures. *Transportation Research Part C: Emerging Technologies*, 18(6):950–962.

Belletti, F., Haziza, D., Gomes, G., and Bayen, A. M. (2017). Expert level control of ramp metering based on multi-task deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1198–1207.

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716.

Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

Berrebi, S. J., Hans, E., Chiabaut, N., Laval, J. A., Leclercq, L., and Watkins, K. E. (2018). Comparing bus holding methods with and without real-time predictions. *Transportation Research Part C: Emerging Technologies*, 87:197–211.

Boutilier, C., Cohen, A., Hassidim, A., Mansour, Y., Meshi, O., Mladenov, M., and Schuurmans, D. (2018). Planning and learning with stochastic action sets. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4674–4682.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Cao, Z., Guo, H., Zhang, J., Oliehoek, F., and Fastenrath, U. (2017). Maximizing the probability of arriving on time: A practical q-learning method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 4481–4487.

Cao, Z., Yang, D., Xu, S., Peng, H., Li, B., Feng, S., and Zhao, D. (2020). Highway exiting planner for automated vehicles using reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):990–1000.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

Chandak, Y., Theocharous, G., Metevier, B., and Thomas, P. (2020). Reinforcement learning when all actions are not always available. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3381–3388.

Chen, C., Huang, Y., Lam, W., Pan, T., Hsu, S., Sumalee, A., and Zhong, R. (2022). Data efficient reinforcement learning and adaptive optimal perimeter control of network traffic dynamics. *Transportation Research Part C: Emerging Technologies*, 142:103759.

Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., and Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3414–3421.

Chen, H., Jiao, Y., Qin, Z., Tang, X., Li, H., An, B., Zhu, H., and Ye, J. (2019). Inbede: Integrating contextual bandit with td learning for joint pricing and dispatch of ride-hailing platforms. In *IEEE International Conference on Data Mining*, pages 61–70. IEEE.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34:15084–15097.

Chen, S.-Y., Yu, Y., Da, Q., Tan, J., Huang, H.-K., and Tang, H.-H. (2018). Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1187–1196.

Chen, W., Zhou, K., and Chen, C. (2016). Real-time bus holding control on a transit corridor based on multi-agent reinforcement learning. In *IEEE 19th International Conference on Intelligent Transportation Systems*, pages 100–106. IEEE.

Chu, T., Wang, J., Codecà, L., and Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095.

Dai, Z., Liu, X. C., Chen, Z., Guo, R., and Ma, X. (2019). A predictive headway-based bus-holding strategy with dynamic control point selection: A cooperative game theory approach. *Transportation Research Part B: Methodological*, 125:29–51.

Darmoul, S., Elkosantini, S., Louati, A., and Said, L. B. (2017). Multi-agent immune networks to control interrupted flow at signalized intersections. *Transportation Research Part C: Emerging Technologies*, 82:290–313.

Desjardins, C. and Chaib-Draa, B. (2011). Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1248–1260.

Devailly, F.-X., Larocque, D., and Charlin, L. (2021). Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12.

Devarasetty, P. C., Burris, M., Arthur Jr, W., McDonald, J., and Muñoz, G. J. (2014). Can psychological variables help predict the use of priced managed lanes? *Transportation Research Part F: Traffic Psychology and Behaviour*, 22:25–38.

El-Tantawy, S., Abdulhai, B., and Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1140–1150.

El-Tantawy, S., Abdulhai, B., and Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 18(3):227–245.

Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556.

Farazi, N. P., Zou, B., Ahamed, T., and Barua, L. (2021). Deep reinforcement learning in transportation research: A review. *Transportation Research Interdisciplinary Perspectives*, 11:100425.

Fares, A. and Gomaa, W. (2014). Freeway ramp-metering control based on reinforcement learning. In *11th IEEE International Conference on Control & Automation*, pages 1226–1231. IEEE.

Finn, C. and Levine, S. (2018). Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *International Conference on Learning Representations*.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2017). Noisy networks for exploration. *CoRR*, abs/1706.10295.

Fujimoto, S., Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

Gao, W., Gao, J., Ozbay, K., and Jiang, Z.-P. (2019). Reinforcement-learning-based cooperative adaptive cruise control of buses in the lincoln tunnel corridor with time-varying topology. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3796–3805.

Guo, Q., Angah, O., Liu, Z., and Ban, X. J. (2021). Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transportation Research Part C: Emerging Technologies*, 124:102980.

Gupta, J. K., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer.

Haydari, A. and Yilmaz, Y. (2022). Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):11–32.

He, J., Yang, H., Tang, T.-Q., and Huang, H.-J. (2018). An optimal charging station location model with the consideration of electric vehicle's driving range. *Transportation Research Part C: Emerging Technologies*, 86:641–654.

He, S. and Shin, K. G. (2019). Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In *The World Wide Web Conference*, pages 2806–2813.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017). Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 1480–1490. PMLR.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Holler, J., Vuorio, R., Qin, Z., Tang, X., Jiao, Y., Jin, T., Singh, S., Wang, C., and Ye, J. (2019). Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In *IEEE International Conference on Data Mining*, pages 1090–1095. IEEE.

James, J. and Lam, A. Y. (2017). Autonomous vehicle logistic system: Joint routing and charging strategy. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2175–2187.

James, J., Yu, W., and Gu, J. (2019). Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3806–3817.

Jiang, Z., Fan, W., Liu, W., Zhu, B., and Gu, J. (2018). Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours. *Transportation Research Part C: Emerging Technologies*, 88:1–16.

Jin, J., Zhou, M., Zhang, W., Li, M., Guo, Z., Qin, Z., Jiao, Y., Tang, X., Wang, C., Wang, J., et al. (2019). Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1983–1992.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Ke, J., Cen, X., Yang, H., Chen, X., and Ye, J. (2019). Modelling drivers' working and recharging schedules in a ride-sourcing market with electric vehicles and gasoline vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 125:160–180.

Khadilkar, H. (2018). A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):727–736.

Kim, D., Ko, J., and Park, Y. (2015). Factors affecting electric vehicle sharing program participants' attitudes about car ownership and program participation. *Transportation Research Part D: Transport and Environment*, 36:96–106.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., and Pérez, P. (2022). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926.

Kok, J. R. and Vlassis, N. (2005). Using the max-plus algorithm for multiagent decision making in coordination graphs. In *Robot Soccer World Cup*, pages 1–12. Springer.

Kok, J. R. and Vlassis, N. (2006). Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7(65):1789–1828.

Lanzaro, G., Sayed, T., and Alsaleh, R. (2022). Can motorcyclist behavior in traffic conflicts be modeled? a deep reinforcement learning approach for motorcycle-pedestrian interactions. *Transportmetrica B: Transport Dynamics*, 10(1):396–420.

Laporte, G., Mesa, J. A., and Perea, F. (2010). A game theoretic framework for the robust railway transit network design problem. *Transportation Research Part B: Methodological*, 44(4):447–459.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Lewis, F. L., Vrabie, D., and Syrmos, V. L. (2012). *Optimal control*. John Wiley & Sons.

Li, C., Bai, L., Liu, W., Yao, L., and Waller, S. T. (2021a). Urban mobility analytics: A deep spatial–temporal product neural network for traveler attributes inference. *Transportation Research Part C: Emerging Technologies*, 124:102921.

Li, G. and Görges, D. (2019). Ecological adaptive cruise control for vehicles with step-gear transmission based on reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4895–4905.

Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G., and Ye, J. (2019). Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, pages 983–994.

Li, Y., Zheng, Y., and Yang, Q. (2018). Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1724–1733.

Li, Z., Chu, T., Kolmanovsky, I. V., and Yin, X. (2017a). Training drift counteraction optimal control policies using reinforcement learning: An adaptive cruise control example. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2903–2912.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W. (2017b). Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3204–3217.

Li, Z., Yu, H., Zhang, G., Dong, S., and Xu, C.-Z. (2021b). Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 125:103059.

Lian, R., Peng, J., Wu, Y., Tan, H., and Zhang, H. (2020). Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle. *Energy*, 197:117297.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations*.

Lim, S., Sommer, C., Nikolova, E., and Rus, D. (2013). Practical route planning under delay uncertainty: Stochastic shortest path queries. In *Robotics: Science and Systems*, volume 8, pages 249–256. MIT Press.

Lin, B., Ghaddar, B., and Nathwani, J. (2021). Deep reinforcement learning for the electric vehicle routing problem with time windows. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11.

Lin, K., Zhao, R., Xu, Z., and Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1774–1783.

Liu, M., Zhao, F., Niu, J., and Liu, Y. (2021). Reinforcementdriving: Exploring trajectories and navigation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):808–820.

Liu, T., Hu, X., Li, S. E., and Cao, D. (2017). Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle. *IEEE/ASME Transactions on Mechatronics*, 22(4):1497–1507.

Liu, T., Zou, Y., Liu, D., and Sun, F. (2015a). Reinforcement learning–based energy management strategy for a hybrid electric tracked vehicle. *Energies*, 8(7):7243–7260.

Liu, W., Yin, Y., and Yang, H. (2015b). Effectiveness of variable speed limits considering commuters' long-term response. *Transportation Research Part B: Methodological*, 81:498–519.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *21st International Conference on Intelligent Transportation Systems*, pages 2575–2582. IEEE.

Lou, K., Yang, Y., Wang, E., Liu, Z., Baker, T., and Bashir, A. K. (2020). Reinforcement learning based advertising strategy using crowdsensing vehicular data. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4635–4647.

Lowe, R., WU, Y., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30:6379–6390.

Luo, M., Zhang, W., Song, T., Li, K., Zhu, H., Du, B., and Wen, H. (2020). Rebalancing expanding EV sharing systems with deep reinforcement learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 1338–1344.

Manchella, K., Umrawal, A. K., and Aggarwal, V. (2021). Flexpool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation. *IEEE Transactions on Intelligent Transportation Systems*, 22(4):2035–2047.

Mannion, P., Duggan, J., and Howley, E. (2015). Parallel reinforcement learning for traffic signal control. *Procedia Computer Science*, 52:956–961.

Mannion, P., Duggan, J., and Howley, E. (2016). An experimental review of reinforcement learning algorithms for adaptive traffic signal control. *Autonomic Road Transport Support Systems*, pages 47–66.

Mao, C., Liu, Y., and Shen, Z.-J. M. (2020). Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 115:102626.

Mao, C. and Shen, Z. (2018). A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. *Transportation Research Part C: Emerging Technologies*, 93:179–197.

Mao, H., Alizadeh, M., Menache, I., and Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pages 50–56.

Markov, A. A. (1954). Theory of algorithms. Springer.

Menda, K., Chen, Y.-C., Grana, J., Bono, J. W., Tracey, B. D., Kochenderfer, M. J., and Wolpert, D. (2018). Deep reinforcement learning for event-driven multi-agent decision processes. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1259–1268.

Minsky, M. L. (1954). *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Princeton University.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Mo, K., Zhang, Y., Li, S., Li, J., and Yang, Q. (2018). Personalizing a dialogue system with transfer reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mousavi, S. S., Schukat, M., and Howley, E. (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7):417–423.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

Nascimento, E. R., Bajcsy, R., Gregor, M., Huang, I., Villegas, I., and Kurillo, G. (2021). On the development of an acoustic-driven method to improve driver's comfort based on deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2923–2932.

Nguyen, D. T., Kumar, A., and Lau, H. C. (2017). Policy gradient with value function approximation for collective multiagent planning. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4320–4330.

Ni, W. and Cassidy, M. J. (2019). Cordon control with spatially-varying metering rates: A reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 98:358–369.

Nishi, T., Otaki, K., Hayakawa, K., and Yoshimura, T. (2018). Traffic signal control based on reinforcement learning with graph convolutional neural nets. In *21st International Conference on Intelligent Transportation Systems*, pages 877–883. IEEE.

Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., Bazzan, A. L., and Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, page 116830.

Oda, T. and Joe-Wong, C. (2018). Movi: A model-free approach to dynamic fleet management. In *IEEE INFOCOM Conference on Computer Communications*, pages 2708–2716. IEEE.

Ono, N. and Fukumoto, K. (1996). Multi-agent reinforcement learning: A modular approach. In *2nd International Conference on Multiagent Systems*, pages 252–258.

Ozan, C., Baskan, O., Haldenbilen, S., and Ceylan, H. (2015). A modified reinforcement learning algorithm for solving coordinated signalized networks. *Transportation Research Part C: Emerging Technologies*, 54:40–55.

Pan, L., Cai, Q., Fang, Z., Tang, P., and Huang, L. (2019). A deep reinforcement learning framework for rebalancing dockless bike sharing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1393–1400.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pan, T., Guo, R., Lam, W. H., Zhong, R., Wang, W., and He, B. (2021). Integrated optimal control strategies for freeway traffic mixed with connected automated vehicles: A model-based reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 123:102987.

Pandey, V. and Boyles, S. D. (2018). Multiagent reinforcement learning algorithm for distributed dynamic pricing of managed lanes. In *21st International Conference on Intelligent Transportation Systems*, pages 2346–2351. IEEE.

Pandey, V., Wang, E., and Boyles, S. D. (2020). Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. *Transportation Research Part C: Emerging Technologies*, 119:102715.

Pedersen, T. H. and Zacharov, N. (2008). How many psycho-acoustic attributes are needed. *Journal of the Acoustical Society of America*, 123(5):3163–3163.

Prashanth, L. and Bhatnagar, S. (2010). Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):412–421.

Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., and Barth, M. (2019). Deep reinforcement learning enabled self-learning control for energy efficient driving. *Transportation Research Part C: Emerging Technologies*, 99:67–81.

Qi, X., Wu, G., Boriboonsomsin, K., Barth, M. J., and Gonder, J. (2016). Data-driven reinforcement learning–based real-time energy management system for plug-in hybrid electric vehicles. *Transportation Research Record*, 2572(1):1–8.

Qin, Z. T., Zhu, H., and Ye, J. (2022). Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144:103852.

Ramos, G. d. O., 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*, 88:257–271.

Reyad, P. and Sayed, T. (2022). Real-time multi-objective optimization of safety and mobility at signalized intersections. *Transportmetrica B: Transport Dynamics*, pages 1–22.

Rezaee, K., Abdulhai, B., and Abdelgawad, H. (2012). Application of reinforcement learning with continuous state space to ramp metering in real-world conditions. In *The 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1590–1595. IEEE.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3859–3869.

Sathyan, A., Ma, J., and Cohen, K. (2021). Decentralized cooperative driving automation: a reinforcement learning framework using genetic fuzzy systems. *Transportmetrica B: Transport Dynamics*, 9(1):775–797.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Šemrov, D., Marsetič, R., Žura, M., Todorovski, L., and Srdic, A. (2016). Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, 86:250–267.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer.

Shi, J., Gao, Y., Wang, W., Yu, N., and Ioannou, P. A. (2019). Operating electric vehicle fleet for ride-hailing services with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4822–4834.

Shou, Z. and Di, X. (2020). Reward design for driver repositioning using multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 119:102738.

Silva, M. A. L., de Souza, S. R., Souza, M. J. F., and Bazzan, A. L. C. (2019). A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Systems with Applications*, 131:148–171.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395.

PMLR.

Srouji, M., Zhang, J., and Salakhutdinov, R. (2018). Structured control nets for deep reinforcement learning. In *International Conference on Machine Learning*, pages 4742–4751. PMLR.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, pages 1038–1044.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063.

Szita, I. and Lörincz, A. (2006). Learning tetris using the noisy cross-entropy method. *Neural Computation*, 18(12):2936–2941.

Tang, X., Li, M., Lin, X., and He, F. (2020). Online operations of automated electric taxi fleets: An advisor-student reinforcement learning framework. *Transportation Research Part C: Emerging Technologies*, 121:102844.

Tang, X., Qin, Z., Zhang, F., Wang, Z., Xu, Z., Ma, Y., Zhu, H., and Ye, J. (2019). A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Thrun, S. and Schwartz, A. (1993). Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum*, volume 6, pages 1–9.

Tom, S., John, Q., Ioannis, A., and David, S. (2016). Prioritized experience replay. *The International Conference on Learning Representations (Poster)*.

Tumer, K. and Agogino, A. (2007). Distributed agent-based air traffic flow management. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8.

Van der Pol, E. and Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems*.

Van Hasselt, H. (2010). Double q-learning. *Advances in Neural Information Processing Systems*, 23.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195.

Wachi, A. (2019). Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6006–6012.

Wan, Z., Li, H., He, H., and Prokhorov, D. (2018). Model-free real-time ev charging scheduling based on deep reinforcement learning. *IEEE Transactions on Smart Grid*, 10(5):5246–5257.

Wang, J. and Sun, L. (2020). Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework. *Transportation Research Part C: Emerging Technologies*, 116:102661.

Wang, T., Cao, J., and Hussain, A. (2021a). Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 125:103046.

Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.-F., Wang, W. Y., and Zhang, L. (2019). Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638.

Wang, Y., Li, M., Lin, X., and He, F. (2021b). Online operations strategies for automated multistory parking facilities. *Transportation Research Part E: Logistics and Transportation Review*, 145:102135.

Wang, Z., Qin, Z., Tang, X., Ye, J., and Zhu, H. (2018). Deep reinforcement learning with knowledge transfer for online rides order dispatching. In *2018 IEEE International Conference on Data Mining*, pages 617–626. IEEE.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Wegener, M., Koch, L., Eisenbarth, M., and Andert, J. (2021). Automated eco-driving in urban scenarios using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 126:102967.

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., and Li, Z. (2019a). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1290–1298.

Wei, H., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., Zhang, W., Zhu, Y., Xu, K., and Li, Z. (2019b). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922.

Wei, H., Zheng, G., Yao, H., and Li, Z. (2018). Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505.

Wei, Y., Mao, M., Zhao, X., Zou, J., and An, P. (2020). City metro network expansion with reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2646–2656.

Wiering, M., Vreeken, J., Van Veenen, J., and Koopman, A. (2004). Simulation and optimization of traffic in a city. In *IEEE Intelligent Vehicles Symposium*, pages 453–458. IEEE.

Wiering, M. A. (2000). Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the 17th International Conference*, pages 1151–1158.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Wu, J., Wei, Z., Liu, K., Quan, Z., and Li, Y. (2020a). Battery-involved energy management for hybrid electric bus based on expert-assistance deep deterministic policy gradient algorithm. *IEEE Transactions on Vehicular Technology*, 69(11):12786–12796.

Wu, Y., Tan, H., Peng, J., Zhang, H., and He, H. (2019). Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus. *Applied Energy*, 247:454–466.

Wu, Y., Tan, H., Qin, L., and Ran, B. (2020b). Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm. *Transportation Research Part C: Emerging Technologies*, 117:102649.

Xu, B., Wang, Y., Wang, Z., Jia, H., and Lu, Z. (2021). Hierarchically and cooperatively learning traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 669–677.

Xu, N., Zheng, G., Xu, K., Zhu, Y., and Li, Z. (2019). Targeted knowledge transfer for learning traffic signal plans. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 175–187. Springer.

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., Liu, C., Bian, W., and Ye, J. (2018). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 905–913.

Yang, K., Zheng, N., and Menendez, M. (2017). Multi-scale perimeter control approach in a connected-vehicle environment. *Transportation Research Procedia*, 23:101–120.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. (2018). Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5571–5580. PMLR.

Yang, Z., Zhu, F., and Lin, F. (2021). Deep-reinforcement-learning-based energy management strategy for supercapacitor energy storage systems in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):1150–1160.

Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., and Komisarczuk, P. (2017). A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys*, 50(3):1–38.

Ye, Y., Zhang, X., and Sun, J. (2019). Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. *Transportation Research Part C: Emerging Technologies*, 107:155–170.

Yin, J., Chen, D., and Li, L. (2014). Intelligent train operation algorithms for subway by expert system and reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2561–2571.

Ying, C.-s., Chow, A. H., and Chin, K.-S. (2020). An actor-critic deep reinforcement learning approach for metro train scheduling with rolling stock circulation under stochastic demand. *Transportation Research Part B: Methodological*, 140:210–235.

Yinlong, Y., Zhuliang, Y., Zhenghui, G., Yao, Y., Wu, W., Xiaoyan, D., Jingcong, L., and Yuanqing, L. (2019). A novel multi-step q-learning method to improve data efficiency for deep reinforcement learning. *Knowledge-Based Systems*, 175:107–117.

Yoon, J., Kim, S., Byon, Y.-J., and Yeo, H. (2020). Design of reinforcement learning for perimeter control using network transmission model based macroscopic traffic simulation. *Plos One*, 15(7):e0236655.

Yu, Z., Liang, S., Wei, L., Jin, Z., Huang, J., Cai, D., He, X., and Hua, X.-S. (2020). Macar: Urban traffic light control via active multi-agent communication and action rectification. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 2491–2497.

Zang, X., Yao, H., Zheng, G., Xu, N., Xu, K., and Li, Z. (2020). Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1153–1160.

Zhang, C., Liu, Y., Wu, F., Tang, B., and Fan, W. (2021a). Effective charging planning based on deep reinforcement learning for electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):542–554.

Zhang, E. and Masoud, N. (2021). Increasing gps localization accuracy with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2615–2626.

Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., and Li, Z. (2019a). Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*, pages 3620–3624.

Zhang, H., Liu, C., Zhang, W., Zheng, G., and Yu, Y. (2020a). Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1783–1792.

Zhang, K., He, F., Zhang, Z., Lin, X., and Li, M. (2020b). Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 121:102861.

Zhang, P., Xiong, L., Yu, Z., Fang, P., Yan, S., Yao, J., and Zhou, Y. (2019b). Reinforcement learning-based end-to-end parking for automatic parking system. *Sensors*, 19(18):3996.

Zhang, Q., Wu, K., and Shi, Y. (2020c). Route planning and power management for phevs with reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(5):4751–4762.

Zhang, R., Ishikawa, A., Wang, W., Striner, B., and Tonguz, O. K. (2021b). Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):404–415.

Zhao, S., Yang, H., and Wu, Y. (2021). An integrated approach of train scheduling and rolling stock circulation with skip-stopping pattern for urban rail transit lines. *Transportation Research*

Part C: Emerging Technologies, 128:103170.

Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., Li, Y., Xu, K., and Li, Z. (2019). Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1963–1972.

Zhou, D. and Gayah, V. V. (2021a). Model-free perimeter metering control for two-region urban networks using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 124:102949.

Zhou, D. and Gayah, V. V. (2021b). Model free perimeter metering control for urban networks using deep reinforcement learning. In *100th Annual Meeting of the Transportation Research Board*.

Zhou, M., Jin, J., Zhang, W., Qin, Z., Jiao, Y., Wang, C., Wu, G., Yu, Y., and Ye, J. (2019a). Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2645–2653.

Zhou, M., Yu, Y., and Qu, X. (2019b). Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: a reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):433–443.

Zhu, F. and Ukkusuri, S. V. (2014). Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 41:30–47.

Zhu, M., Wang, X., and Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 97:348–368.

Zhu, M., Wang, Y., Pu, Z., Hu, J., Wang, X., and Ke, R. (2020). Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies*, 117:102662.

Zhu, Z. and Zhao, H. (2021). A survey of deep rl and il for autonomous driving policy learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14043–14065.