CTRL: Cooperative Traffic Tolling via Reinforcement Learning

Yiheng Wang
yhwang0828@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Hexi Jin
sjtujinhexi@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Guanjie Zheng*
gjzheng@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

People have been working long to tackle the traffic congestion problem. Among the different measures, traffic tolling has been recognized as an effective way to mitigate citywide congestion. However, traditional tolling methods can not deal with the dynamic traffic flow in cities. Meanwhile, thanks to the development of traffic sensing technology, how to set appropriate dynamic tolling according to real-time traffic observations has attracted research attention in recent years.

In this paper, we put the dynamic tolling problem in a reinforcement learning setting and try to tackle the three key challenges of complex state representation, pricing action credit assignment, and route price relative competition. We propose a soft actor-critic method with (1) a route-level state attention, (2) an interpretable and provable reward design, and (3) a competition-aware Q attention. Extensive experiments on real datasets have shown the superior performance of our proposed method. In addition, interesting analysis on pricing actions and vehicle routes have demonstrated why the proposed method can outperform baselines.

CCS CONCEPTS

• **Information systems** → *Data mining*; *Spatial-temporal systems.*

KEYWORDS

dynamic tolling; reinforcement learning

ACM Reference Format:
Yiheng Wang, Hexi Jin, and Guanjie Zheng. 2022. CTRL: Cooperative Traffic Tolling via Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557112

1 INTRODUCTION

Traffic congestion has become a serious problem for many cities due to the rapid urbanization and corresponding explosive increase in vehicle registration numbers. This has correspondingly caused long commuting times, reduced quality of life, increased energy consumption, and urban ecological degradation [31]. Therefore, various measures have been investigated to tackle traffic congestion, including intelligent traffic signal control [10, 26], road restrictions [15], and so on. Compared with other measures that usually only cover local regions, traffic tolling [2] is expected to provide city-level impact on the traffic pattern.

Traffic tolling aims to assign a price to each road (or route) according to its congestion condition. Thus, drivers are motivated to choose a more economical route (usually a less-congested route). In this way, vehicles are distributed among different routes and congestion is alleviated.

Current tolling methods can be classified into two categories: static tolling and dynamic tolling. Static tolling [28, 32] sets tolls merely based on historical data, which fails to adapt well to the changing traffic volume. In terms of dynamic tolling, many works start developing a pricing strategy according to the real-time traffic conditions. Previous dynamic pricing methods usually set tolls based on observations on roads, e.g., delay increase [22] and social welfare measures [30]. However, these methods are based on unrealistic assumptions which are difficult to satisfy and the parameters in these methods need to be tuned repeatedly under different roadnet and traffic flow. Recently, reinforcement learning methods are becoming popular [5, 11, 17–20]. These methods can directly optimize towards the objective via a pre-defined reward function [2]. Hence, they can adapt to different roadnet and traffic flows via a learning process.

However, applying reinforcement learning to develop a reasonable dynamic tolling strategy faces several key challenges.

- **State representation**. How to accurately represent the road congestion conditions is still an open question, given the complex road network connection and traffic flow interaction. (1) Previous RL approaches to traffic tolling [19, 20] often simply take the number of vehicles on a road as the state. This alone cannot accurately reflect the congestion because the number of vehicles and the degree of congestion are not directly related (if a lot of vehicles proceed smoothly, this should not be a problem concerning congestion). (2) The distribution of vehicles on their routes matters. As shown in Fig. 1 (a), when other conditions are the same (e.g., road length, congested road number, vehicle volume) while the traffic flow distribution is different, Route 1 is more likely to cause congestion diffusion than Route 3 (more vehicles gather in certain consecutive roads in Route 1). Due to the connectivity of the roadnet, these interactions among roads and their upstream or downstream should be captured.
- **Credit assignment**. The interwoven road network makes it difficult to tell which price is responsible for a resulting traffic efficiency change. This is because the results of multiple route choices are mixed. Here, we take the metric throughput (TP) as an example, since it is frequently used in traffic tolling studies [18–20]. It remains challenging to answer to what extent each pricing

[*] Guanjie Zheng is the corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557112

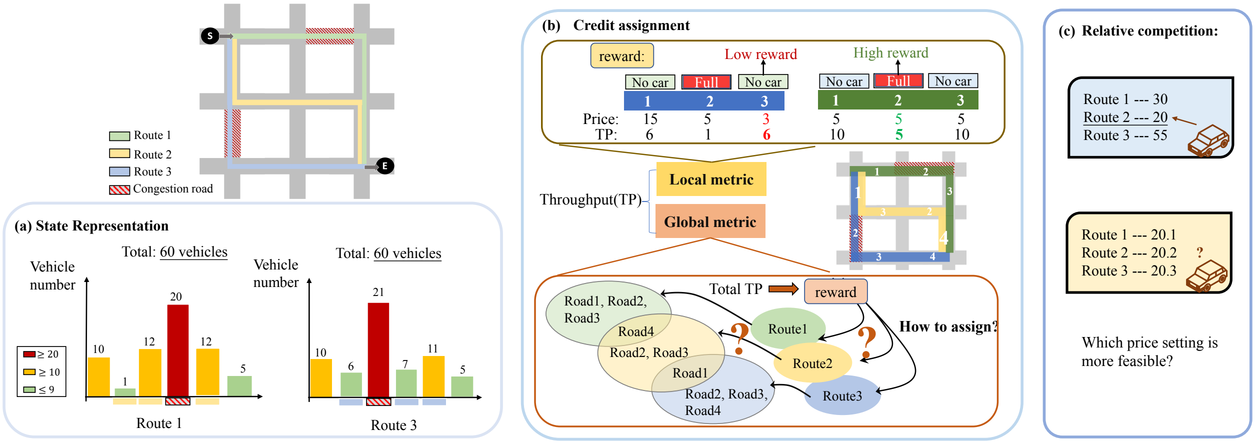

Figure 1: An example demonstration of the dynamic traffic tolling problem setting and three major challenges.

and routing action contribute to the final throughput number. As shown in Fig. 1 (b), when assigning credit to local roads, road 3 in the blue route may get a low reward because its congested upstream road are doing the wrong thing: congested road 2 with low throughput is charging a low price and empty road 1 is charging too much. On the contrary, road 2 in the green route may get a high reward due to its neighbor having set a correct price and support high throughput, even though road 2 itself is doing a poor job. Moreover, when trying to assign a global metric to each route, throughput over time is hard to be assigned in such a multi-agent system [4, 24], since the traffic flow from roads to roads overtime, and the decomposition relationship between route traffic and road traffic vary over time.

- **Relative competition**. When facing route choices, drivers will choose the route at a *relatively low* price. Hence, it is crucial to learn the price of competing routes in a nutshell, rather than separately. For instance, as shown in Fig. 1 (c), these two sets of route pricing will result in the same choice of route 2. However, which one is more feasible? Two realistic issues need to be considered. (1) The relative price comparison among routes should be sharp, so that drivers will choose the route designed by the algorithm with no hesitation. (2) The total price of a route should stick to a reasonable range and prices among different OD pairs should remain feasible. Otherwise, people on different trips may complain. In contrast, most of the previous studies set the absolute price for each road or route independently and therefore lead to the failure to learn the optimal strategy.

To address these challenges, we propose a novel route-based method for dynamic tolling named **C**ooperative traffic **T**olling via **R**einforcement **L**earning (CTRL). To appropriately represent the congestion level of each route, we propose a state-based attention mechanism that can aggregate the observations from roads considering the traffic distribution. To address the challenge of pricing credit assignment, we derive a proper reward function that can be decomposed from the global objective to road level measurements. This guarantees that optimizing the reward values is equivalent to optimizing the global objective. As for the modeling of relationships between a group of routes, we re-designed the Q-network of the RL algorithm with a competition-aware Q-attention mechanism. Extensive experiments are conducted on the real-world road network of three cities. The results show that our approach can significantly outperform the state-of-the-art methods on each metric.

In summary, the contributions can be summarized as follows.

- We propose a novel method of dynamic tolling that solves the complex state representation, pricing credit assignment and route price relative competition challenge.
- The reward function is derived from the decomposition of the global objective and hence can guarantee performance.
- We conduct extensive experiments on a widely used microscopic traffic simulator. This is an essential step before applying it to the real world, compared with previous mathematical model simulations. Experimental results on real-world datasets show that this design has practical value for road congestion improvement.

2 RELATED WORK

Road pricing is a measure designed to reduce delays and congestion by charging drivers on roads. Compared to traffic light control, road pricing may intervene earlier in the event of potential congestion and influence drivers' route choices in advance. The development of road tolling models can be divided into three categories.

Static tolling Most of the static road pricing model is based on the analysis of historical road system data. Yang and Zhang adopt a genetic algorithm to search for optimal toll locations and simulate an annealing method to determine toll levels. [28]. Joksimovic et al. model the pricing problem as a bi-level optimization. They use a heuristic search algorithm to find the optimal toll pattern [12]. Zhou et al. propose a trial-and-error congestion pricing scheme applying the two-level iteration [32]. These methods are usually based on mathematical models with ideal assumptions, and since they assume fixed traffic flow, they cannot fully utilize the charging mechanism to control the flow of traffic in real-time.

Initial dynamic tolling Then, researchers start to design dynamic tolling models. Zhang et al. [30] consider road pricing as a multi-player game, and they propose a pricing model among multiple regions by using Stackelberg and Nash games. But it actually ignores the mechanism of congestion by applying regional pricing and a lot of assumptions are made. Bui et al. design a novel mechanism User-Centric Dynamic Pricing (UCDP) and they calculate tolls based on marginal cost and tailor paths to heterogeneous users' preferences according to the current traffic condition [3].

Sharon proposes Δ-tolling [22] that computes tolls proportional to the difference between observed and free-flow travel times. Although these tolls are based on real-time traffic flows, the methods both assume the parameters for each road to be the same and still do not consider the impact of other roads between networks. Moreover, the models of these methods usually lack generalization and are not fully applicable to different road networks, requiring additional tuning work.

Reinforcement learning for dynamic tolling Recent studies on road pricing have verified the advantage of Reinforcement learning (RL) in the function approximation of high-dimensional input environment [13, 25]. The tolling problem is formulated as a Markov Decision Problem (MDP) with a continuous and bounded action space. Mirzaei [17] improves Δ-tolling [22] by applying RL method to set distinctive parameters for each road, so as to incentivize self-interested agents to coordinate. Chen [5] develops a Policy Gradient method, PG-β, to allocate road tolls by RL. However, these methods do not consider the dynamic combination of roads in different routes when setting a price for a specific road. Later, Qiu [20] proposes DPG-β via Multi-Agent Deep Reinforcement Learning with Edge-Based Graph Convolutional Network (MARL-eGCN), which improves the performance of PG-β by employing deep neural networks and speeds up target value update by employing temporal difference. They also employ GCN to extract the spatial correlation within the roadnet. Unfortunately, the methods above still use domain knowledge such as manually designing an agent-wise reward function which do not resolve the credit assignment problem in a multi-agent environment.

3 PRELIMINARIES

We first introduce the notations, concepts, and the problem formulation.

3.1 Notations and Concepts

To begin with, we introduce some necessary concepts in our problem.

- **Roadnet:** The roadnet is represented by a graph $G = (V, E, X)$, where vertices V are roads (each road contains three lanes: straight, left, and right), edges E are intersections representing the connection between roads, X are features of roads in the roadnet, as shown in Fig. 2. Note that vehicles in different lanes are separately considered in the model. However, to avoid unnecessary confusion, we will only use the term "road" rather than "lane" in the following sections. Thus, the feature of each route can be obtained by applying the route adjacency matrix on the graph.
- **Traffic flow:** The traffic in the system is composed of multiple traffic flows. Each traffic flow is defined as the group of vehicles sharing the same origin-destination (OD) pairs. Vehicles depart the origin and head for the destination at a specified time. They can change their routes according to traffic conditions.
- **Tolling:** Each road will be assigned a price a. Drivers can obtain the total price of a route by adding up the prices of the roads and making route choices correspondingly.

Table 1: Notations and descriptions

Notation	Descriptions
G	Road Network Graph
V, E	Vertex set (roads) and edge set (intersections) of G
X	Feature of roads in G
i	Road id
a	Tolling price
v^{max}	Max speed of all vehicles
d_{global}	Global delay
D	OD pair
N	Number of all vehicles on the roadnet
N_i	Number of vehicles on the road i
$d_{i,j}^n$	Delay of vehicle j on the road i at time step n
d_i^n	Delay of the road i at time step n
$t_{i,j}^{\langle 0 \rangle}$	Free flow travel time of vehicle j on the road i
t_i^n	Average travel time of the road i from time step $n-1$ to n
$R_{i,j}^n$	Traveling distance of vehicle j on the road i from time step $n-1$ to n

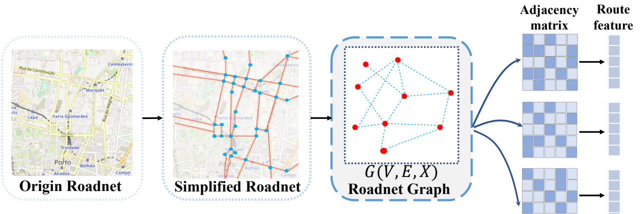

Figure 2: A graph representation of the traffic system.

- **Route choice model:** Each vehicle can choose its route with the lowest total cost among all the alternative routes. We assume that vehicles do not turn around and all the drivers are homogeneous. The route cost is calculated as $\sum_{i=1}^{M_r} a_i$, where M_r is the number of roads in the route r and a_i is the price of the i^{th} road in the route.

3.2 Problem Definition

The tolling problem can be formulated as a Markov Decision Process (MDP). The objective is to learn a policy to set a price for each route, which could optimize the average travel time and throughput of the network. The problem can be formally defined as follows:

Problem 1. *Given a city roadnet represented by $G = (V, E, D, X)$ in which V is the set of roads, E is the set of intersections connecting roads, D is the set of traffic origin-destination (OD) pairs, and X is the feature of each road. For every OD, there exist several alternative routes. Vehicles choose one route with the lowest price after entering the roadnet. The goal is to derive a joint action a for all roads interfering vehicles' route choices, to minimize the average travel time for all vehicles.*

4 METHOD

In this section, we propose a reinforcement learning model to tackle this tolling problem, which is built upon the famous Soft Actor-Critic algorithm. This method is named **C**ooperative traffic **T**olling via **R**einforcement **L**earning (CTRL), in which traffic observations cooperate to generate the solution. The state, reward and effectiveness of action in previous road-level methods are greatly influenced by the surrounding road, and the individual road can hardly achieve an impact on the driver's route choice. Considering that the price comparison from the driver perspective happens at the route-level, we conduct a derivation in converting the optimization objective into route-level objective. Then, to extract the dynamic and complex spatial interactions among the roads within each route, we propose a novel state representation attention to aggregate the state features into route level. Further, to capture the relative competing relationship between the routes of each origin-destination (OD) pair, we design a Q-attention network to comprehensively consider the values for all candidate routes.

We will first introduce the RL setting and the reward design in Section 4.2. Then we demonstrate how we aggregate the road-level state observations into route-level via Route-Level State Attention in Section 4.3. Further, the route competition-aware attention mechanism is shown in Section 4.4. In the end, we introduce how the proposed attention mechanisms are incorporated into the network design and the training details in Section 4.5.

4.1 Tolling in Reinforcement Learning Setting

In order to solve the dynamic tolling problem, we formulate the problem in the RL setting as follows. Since the observation is usually obtained at road-level while the pricing is carried out at the route level, we will discuss how to formulate the agent from road view to route view.

Road view From the view of roads, we define the following measurements to describe the congestion level and the effect of pricing actions.

- **Observation.** The key observation of a road is the delay index Ω_i^n for the current road at time n. This is defined as the following equation:

$$\Omega_i^n = \zeta(\lambda_i^n - \lambda_i^{\langle 0 \rangle}). \tag{1}$$

$$\lambda_i^n = \frac{\sum_1^{J_i}(n_{i,j,2} - n_{i,j,1})}{J_i} \tag{2}$$

Here, free flow time $\lambda_i^{\langle 0 \rangle} = \frac{L_i}{v^{max}}$ is the time to pass the road i with the maximum speed v^{max}, and L_i is the length of road i. λ_i^n is the actually measured average travel time at time step n on road i, where the vehicles used for calculation are restricted to those that have left the road, as shown in Eq. (2), J_i is the number of vehicles that have left road i, $n_{i,j,2}$ and $n_{i,j,1}$ is the time step that vehicle j leaves and enters the road i respectively. Note that ζ is a constant scaling factor which is set to be 0.8 according to literature. Tuning this coefficient will only change the scale of the features and hence will not affect the result.
- **Action.** The action for each road is defined as the toll price for this road. For the sake of fairness, we define the range of price for each road as a value in $[0, 10]$. This way, the total price of routes can be bounded in a feasible range.
- **Reward.** The reward for each road is defined as the cumulative distance traveled by vehicles on the road i from time step $n-1$ to n. You can obtain the average speed for each road via calculating $R_i^n/\delta t$, where δt is the time step gap between time $n-1$ and n. Since δt is a constant, we omit it to make the following derivation host:

$$R_i^n = \sum_{j=1}^{N_i} R_{i,j}^n. \tag{3}$$

Lemma 1. *For a traffic system represented by $G(V, E)$, minimizing the delay index for all the vehicles in the whole system is equivalent to minimizing the delay index for each road individually. Note that the global delay index is defined as below, N denotes the total vehicle number, T denotes the whole tme scope, and d_j is the total delay of vehicle j:*

$$d_{vehicles} = \sum_{j=1}^{N} d_j = \sum_{j=1}^{N}\sum_{n=1}^{T} d_j^n. \tag{4}$$

Proof. For each time step n, we can calculate the global delay by summing up the delay for each road as follows. Here, V is the set of roads, N_i^n means the set of vehicles that are on the road i during time step n, $t_{i,j}^n$ is the travel time of vehicle j on the road i from step n, Δt is the interval between steps, which is a constant value, and $R_{i,j}^n$ represents the distance that vehicle j proceeds on road i during time step n:

$$\begin{aligned} d_{global}^n &= \sum_{i=1}^{V} d_i^n = \sum_{i=1}^{V}\sum_{j=1}^{N_i^n} d_{i,j}^n \\ &= \sum_{i=1}^{V}\sum_{j=1}^{N_i^n}\left(t_{i,j}^n - t_{i,j}^{\langle 0 \rangle}\right) = \Delta t - \sum_{i=1}^{V}\sum_{j=1}^{N_i^n}\frac{R_{i,j}^n}{v^{max}}. \end{aligned} \tag{5}$$

Thus, the global delay for the whole time scope T can be obtained as follows:

$$\begin{aligned} d_{global} &= \sum_{n=1}^{T}\sum_{i=1}^{V} d_i^n \\ &= \sum_{n=1}^{T}\Delta t - \frac{1}{v^{max}}\sum_{n=1}^{T}\sum_{i=1}^{V}\sum_{j=1}^{N_i^n} R_{i,j}^n. \end{aligned} \tag{6}$$

Note that, the first term adds up to a constant $T \cdot \Delta t$. Then, for the second term, by swapping the order of the summation operator and representing it using the vehicle view, we have the following equation holds:

$$\sum_{n=1}^{T}\sum_{i=1}^{V}\sum_{j=1}^{N_i^n} R_{i,j}^n = \sum_{j=1}^{N}\sum_{n=1}^{T}\sum_{i=1}^{V_j^n} R_{i,j}^n, \tag{7}$$

where V_j^n is number of roads that vehicle j travels through from time step n-1 to n. After that, by substituting Eq. (7) into Eq. (6), we have

$$\begin{aligned} d_{global} &= T \cdot \Delta t - \frac{1}{v^{max}}\sum_{j=1}^{N}\sum_{n=1}^{T}\sum_{i=1}^{V_j^n} R_{i,j}^n \\ &= \sum_{j=1}^{N}\sum_{n=1}^{T}(\Delta t - \frac{R_j^n}{v^{max}}) = \sum_{j=1}^{N}\sum_{n=1}^{T} d_j^n = d_{vehicles}. \end{aligned} \tag{8}$$

Hence, we know that minimizing the delay index for all the vehicles in the whole system is equivalent to minimizing the delay index for each road individually. □

For simplicity, from now on, we use the reward function as the cumulative distance R_i^n.

Route view From the view of routes, we define the following elements.

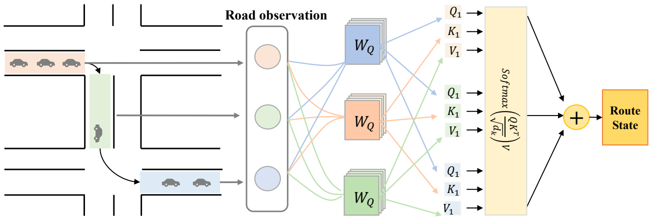

Figure 3: State attention mechanism.

- **State.** The state s_r^n of a route r at time n is an aggregation of observations of roads it covers. We define it as:

$$s_r^n = Agg(\mathbf{d}_i^n), \tag{9}$$

  where $\mathbf{d}_i^n$ is a concatenation of the observations of the roads $i \in r$. The aggregation function Agg can be average function or state attention mentioned in Section 4.3.
- **Action.** The action a_r^n for each route r at time n is the toll price for this route in CTRL bounded in $[0, 10]$. For the method of tolling on roads, the action is the sum of all roads' prices. Since the number of roads in optional routes is comparable, the prices of the routes are also bounded.
- **Reward.** The reward for each route is the average reward of all roads it covers.

We adopt the structure of SAC agent to model our agent [8]. The policy aims to maximize the expected sum of reward over the state-action trajectory distribution of $(\mathbf{a}_n|\mathbf{s}_n)$. The objective with expected entropy over trajectory $\rho_\pi(\mathbf{s}_n)$ is defined as:

$$J(\pi) = \sum_{n=1}^{T} \mathbb{E}_{(\mathbf{s}_n,\mathbf{a}_n)\sim\rho_\pi} \left[r\left(\mathbf{s}_n, \mathbf{a}_n\right) + \alpha\mathcal{H}\left(\pi\left(\cdot \mid \mathbf{s}_n\right)\right)\right], \tag{10}$$

where α is the temperature parameter that determines the weight of entropy term against the reward. This term can effectively control the stochasticity of policy [8].

A tractable policy $\pi_\phi\left(\mathbf{a}_n \mid \mathbf{s}_n\right)$ is considered with the parameter ϕ. The policy network aims to minimize the loss function with the following form:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_n\sim\mathcal{D}} \left[\mathbb{E}_{\mathbf{a}_n\sim\pi_\phi} \left[\alpha \log(\pi_\phi\left(\mathbf{a}_n \mid \mathbf{s}_n\right)) - Q_\theta\left(\mathbf{s}_n, \mathbf{a}_n\right)\right]\right], \tag{11}$$

where D is the data sample, ρ_ϕ is a tractable policy.

It will compute two Q-values and choose the minimum one to stabilize training [8]. The Q-network is modeled as a soft Q-function $(Q_\theta\left(\mathbf{s}_n, \mathbf{a}_n\right))$ whose parameter is θ. It is trained to minimize the soft Bellman residual:

$$\begin{aligned} J_Q(\theta) = \mathbb{E}_{(s_n,a_n)\sim\mathcal{D}} \Big[\frac{1}{2} \Big(& Q_\theta\left(\mathbf{s}_n, \mathbf{a}_n\right) - \left(r\left(\mathbf{s}_n, \mathbf{a}_n\right)\right. \\ & \left. + \gamma\mathbb{E}_{\mathbf{s}_{n+1}\sim p}\left[V_{\bar{\theta}}\left(\mathbf{s}_{n+1}\right)\right]\right) \Big)^2 \Big], \end{aligned} \tag{12}$$

where

$$V_{\bar{\theta}}\left(\mathbf{s}_n\right) = \mathbb{E}_{\mathbf{a}_n\sim\pi} \left[Q_{\bar{\theta}}\left(\mathbf{s}_n, \mathbf{a}_n\right) - \alpha \log \pi\left(\mathbf{a}_n \mid \mathbf{s}_n\right)\right]. \tag{13}$$

And it can be optimized with stochastic gradients [8]:

$$\begin{aligned} \hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta\left(\mathbf{s}_n, \mathbf{a}_n\right) \Big(& Q_\theta\left(\mathbf{s}_n, \mathbf{a}_n\right) - r\left(\mathbf{s}_n, \mathbf{a}_n\right) \\ & -\gamma\left(Q_{\bar{\theta}}\left(\mathbf{s}_{n+1}, \mathbf{a}_{n+1}\right) - \alpha \log(\pi_\phi\left(\mathbf{a}_{n+1} \mid \mathbf{s}_{n+1}\right))\right) \Big). \end{aligned} \tag{14}$$

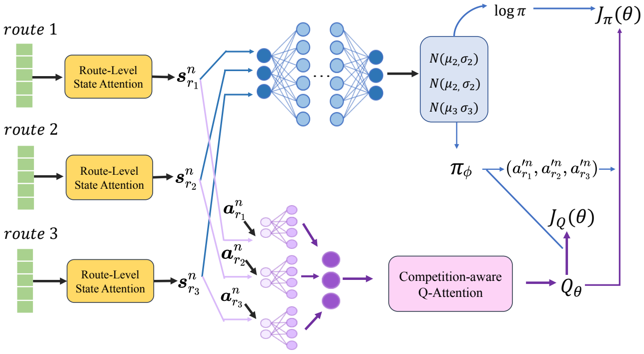

Figure 4: Network architecture of CTRL.

4.2 Route-level State Attention

The state representation should be able to accurately reflect the current road conditions and realize the cooperation of them.

As mentioned earlier, a route of an OD pair consists of multiple roads, and the traffic flows in the upstream and downstream roads will interact with each other. Thus, different traffic distributions may reflect different congestion conditions. Capturing the relationship between such upstream and downstream roads is also an important component of the state representation.

We utilize the attention module for the roads of a route to model the relationship between road sequences and use the output of attention as the state of a route, as shown in Fig. 3. For each route state s_r of route r, state attention calculates s_r as follows:

$$s_r = \frac{1}{M_r} \sum_{i=1}^{M_r} softmax(\frac{(W_s^Q O_r)(W_s^K O_r)^T}{\sqrt{d_k}})(W_s^V O_r), \tag{15}$$

where $O_r = \text{FC}\left([\mathbf{d}_i]\right)$, $\mathbf{d}_i$ is a concatenation of the observations of the roads covered by the route r. M_r is the number of roads covered in route r and d_k is a hyperparameter that controls the projection dimension. W_s^Q, W_s^K and W_s^V are learned linear transformations.

4.3 Competition-aware Q-Attention

The attention mechanism has shown excellent performance in making the model learn to focus on a specific part of the input sequence. In our environment, the prediction of the route score (i.e., Q-value) for each action by a Q-network depends not only on the state and action of a route itself but also on the states and actions of its competing routes. We design an attention-based Q-network to derive the relative Q-values of the routes in a competitive relationship based on their states and actions.

The attention mechanism is designed to calculate the value of state and action for each routes given a pair of ODs, measuring how good the corresponding pricing actions are for a pair of ODs:

$$h_r = \text{FC}\left(\text{ReLU}\left(\text{FC}\left([s_r, a_r]\right)\right)\right), \tag{16}$$

$$Q(\mathbf{s}, \mathbf{a}) = softmax(\frac{(W_q^Q \mathbf{h})(W_q^K \mathbf{h})^T}{\sqrt{d_q}})(W_q^V \mathbf{h}), \tag{17}$$

where $\mathbf{h} = [h_1, h_2, h_3]$, W_q^Q, W_q^K and W_q^V are learned linear transformations. d_q is a hyperparameter that controls the projection dimension. Q-attention network ensures the fairness of the learning policy, the prices between routes are distinguishable and relatively

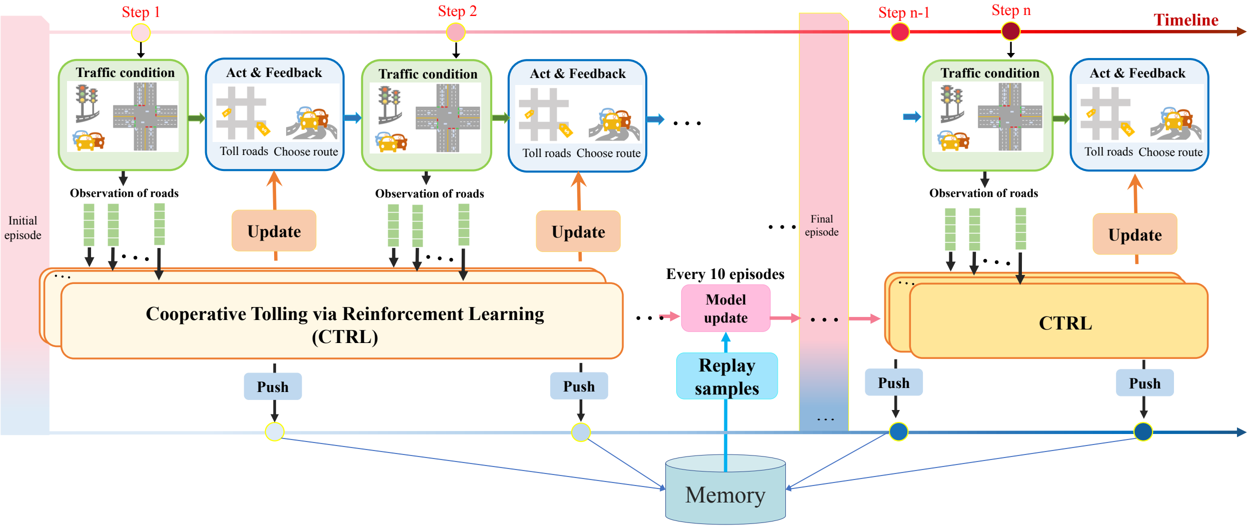

Figure 5: Training pipelines of our CTRL.

fair for all routes on the entire road network, i.e., there is no OD whose routes' prices are much higher or lower than all other routes.

4.4 Implementation details

In this section we will introduce the structure of critic network and actor network and how attention works in it as shown in Fig. 4. In addition, we further show the training pipeline of the proposed method.

4.4.1 Network Models.

Critic Network The network consists of two Q-networks, which compute two Q-values and choose the minimum one to stabilize training, each of which is shown in the down right of Fig. 4. The input to the Q-network is the concatenations of state and action for different routes of an OD pair. Firstly, the input pairs are passed through a hidden layer composed of two linear layers respectively to obtain h_i as in Eq. (16). We derive the Q-value representation of the OD-level for $\mathbf{h}$ from the previously mentioned Q-attention mechanism as in Eq. (17). We also apply target Q-networks Q_{target} to calculate the target Q-values for the stability of the algorithm.

We update the critic by minimizing the TD-loss $\mathcal{L}(\phi)$:

$$\begin{aligned}\mathcal{L}(\phi) &= \frac{1}{2}\left(Q_{\phi}(\mathrm{s},\mathrm{a}) - \hat{Q}_{\phi_{target}}(\mathrm{s},\mathrm{a})\right)^2, \\ \hat{Q}_{\phi_{target}}(\mathrm{s},\mathrm{a}) &= R_D + \gamma\left(\hat{Q}_{\phi_{target}}(\mathrm{s}',\tilde{\mathrm{a}}') - \alpha\log\pi_\theta(\tilde{\mathrm{a}}'|\mathrm{s}')\right),\end{aligned} \tag{18}$$

where $\tilde{\mathbf{a}}' \sim \pi_\theta(\cdot|\mathbf{s}')$, π_θ is the current policy, $\mathbf{s}'$ is next state. R_D is the rewards of OD D:

$$\begin{aligned}R_D &= \sum_{r=1}^{k}\sum_{i=1}^{M_r} R_i \\ &= \sum_{r=1}^{k}\sum_{i=1}^{M_r}\sum_{j=1}^{N_i} R_{ij}\end{aligned} \tag{19}$$

according to the definition of Eq. (3), where k is the number of alternative routes under an OD pair and here k is 3 in our setting.

Target Q-networks is updated with:

$$\phi_{target} \leftarrow \tau\phi_{target} + (1-\tau)\,\phi. \tag{20}$$

Actor Network The actor network takes the state of the OD pair as input and outputs the action distributions and log probabilities of actions for the routes of the OD pair respectively, which can be expressed as $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$. The structure of actor network is shown in the top right of Fig. 4.

Then the actor network is trained by minimizing the loss:

$$\mathcal{L}(\theta) = \alpha\log\pi_\theta(\tilde{\mathrm{a}}|\mathrm{s}) - Q_\phi(\mathrm{s},\tilde{\mathrm{a}}), \tilde{\mathrm{a}} \sim \pi_\theta(\mathrm{a}|\mathrm{s}). \tag{21}$$

The actor network gives the action distribution of each route respectively. In the training phase the action is decided by a non-deterministic policy which we sample in the output normal distribution. In the testing phase, the action is set as the mean of the distribution.

4.4.2 Training Pipelines. As shown in Fig. 5, during the online training, CTRL will interact with the traffic environment in real time and update in the following way:

(1) State representation: In each time step, the observations of each road are extracted from current traffic condition, and each observation is fed into our Route-level State Attention network to generate the route state input of CTRL.

(2) Update: CTRL will make decisions for each agent and provide the current appropriate prices for the driver to choose the route accordingly. The driver will calculate the driving cost of each route and choose the route with the lowest cost.

(3) Push: The agent's decision and subsequent environment's latest feedback will be consolidated into a quadruple $(s_{D_i}^n, a_{D_i}^n, r_{D_i}^n, s_{D_i}^{n+1})$ and stored in memory as experience samples.

(4) Replay samples, Update model: Every 10 episodes, CTRL will use the experience samples stored in the memory to update the whole network.

(5) Repeat step (1)-(4) until the model converges.

Table 2: Overall performance comparison on three real world datasets. Different methods are compared in three measurements, Throughput, ATT (average travel time), and Rewards. Throughput (↑) and Rewards (↑) are the higher the better, while ATT (↓) is the lower the better.

City	Hangzhou			Manhattan			Porto		
Measurement	Throughput	ATT(s)	Rewards	Throughput	ATT(s)	Rewards	Throughput	ATT(s)	Rewards
No-change	1874	1235.14	1464.93	2324	1337.36	2483.97	3849	734.64	4118.63
Random	2406	787.97	1593.72	2683	1156.71	3230.16	4044	631.11	4653.22
Formula	2420	785.50	1094.06	2736	1130.41	3379.64	4051	628.14	4690.09
Δ-tolling	2956	902.02	2040.71	2860	1173.40	3715.68	**4262**	558.92	4905.89
Indi-SAC	2436	895.13	1537.90	2636	1176.50	3143.71	4088	643.55	4884.86
Multi-SAC	2437	935.69	1576.42	2713	1141.11	3339.04	4066	626.25	4833.06
MARL-eGCN	2461	1010.26	1112.30	2708	1143.12	3325.97	4069	642.69	4905.94
CTRL	**3053**	**686.38**	**2760.51**	**2926**	**908.43**	**3780.46**	**4262**	**493.68**	**5337.63**

Table 3: Parameters of our method

Steps	6
Decay for exploration ϵ	0.99
Target smoothing coefficient τ	0.125
Action interval	1800
Target model update interval	120
Batch size	32
Memory size	2000
Episode	500

5 EXPERIMENT

In the experiment, we aim to answer the following research questions.

- **RQ1:** How does CTRL perform on different datasets compared with baselines?
- **RQ2:** How does the state attention module help to improve the performance?
- **RQ3:** Does the Q-Attention module contribute to a fair price?
- **RQ4:** Why can CTRL admit more vehicles than baselines?

5.1 Experimental Settings

5.1.1 Simulation environment. Our experiment uses CityFlow[1] traffic simulator, which is a widely-used multi-agent reinforcement learning environment for large-scale urban traffic scenarios [29].

The simulator takes the road network file and the traffic file as input, and admits vehicles into the simulation system according to the traffic file. Each vehicle will depart its origin and head for its destination at a specific time, via a predefined route (as in the traffic file). Meanwhile, users can access traffic observations (e.g., vehicles on a specific road, the speed of vehicles) from the simulator via Python APIs. In turn, users can also set a new route for each vehicle. To avoid chaos, each vehicle can only reroute once.

5.1.2 Parameter setting. The parameter settings for our traffic environment and experiment are shown in Table 3. Action interval defines the interval between tolling price changes, which is the time of a time step in seconds.

5.1.3 Route selection. Each OD pair in the roadnet has three different alternative routes, i.e., the shortest total distance, the fewest traffic lights and the fewest estimated vehicles. The drivers choose the route from the three candidates with the minimal total price when entering the road network and do not change the route or destination during the travel.

5.2 Dataset

Roadnet We consider three real traffic networks to validate our methods, including Hangzhou dataset of a 4x4 network (16 intersections in total), Manhattan road network of a 3x16 (48 intersections in total) and city-scale Porto roadnet work of 200 roads (70 intersections in total). The former two roadnets are taken from the web page[2] and are wildly used in literature [26, 27], Porto roadnet data is simplified from OpenStreetMap (OSM) data[3].

Traffic flow The traffic flow is obtained from real public city data. Concretely, the traffic flow for Hangzhou and Manhattan is downloaded from the previously mentioned website. These datasets are converted from real traffic camera data and taxi data. The traffic flow for Porto is converted from the taxi dataset used by previous data competition [4] hosted by ECML-PKDD. The traffic flow of each road network contains many OD pairs (Hangzhou has 403 ODs, Manhattan has 818 ODs, and Porto has 817 ODs) and we manually increase the congestion on long-distance routes to observe more obvious effects. Then we choose the relatively congested OD pairs to focus on and control. It is expected that through controlling vehicles on traveling in these OD pairs, all the vehicles inside the system will be influenced.

5.3 Methods for Comparison

- **No change:** No tolls are set, and vehicles follow original routes.
- **Random:** Vehicles choose routes randomly among alternatives.
- **Formula:** The toll of each road p follows the function

$$p = \begin{cases} 0 & x < 5, \\ min(10, \frac{10}{7}x - \frac{50}{7}) & x \geq 5. \end{cases}$$

[1] CityFlow project code can be found at https://cityflow-project.github.io

[2] Roadnet data is provided at https://traffic-signal-control.github.io

[3] OSM is provided at https://www.openstreetmap.org/map=13/41.1603/-8.6385

[4] Data available at https://www.kaggle.com/datasets/crailtap/taxi-trajectory

Here, x is the number of vehicles on each road. We conduct experiments with various values of x and achieve the best performance when x is 5.

- Δ**-tolling [22]** is a dynamic tolling model. Each road's toll follows the function

$$\tau^n = (1 - C)\tau^{n-1} + C\beta(T^{n-1} - T^0) \tag{22}$$

where T^0 is the free-flow travel time, T^{n-1} is the average travel time at time step n, C and β are tuning parameters. C is a decay.
- **Indi-SAC [7]:** This method adopts the soft actor-critic framework and utilizes the same state, action and rewards with our method. Each road agent uses an independent SAC model.
- **Share-SAC:** All road agents share one SAC model. The state, action and rewards are the same as our method.
- **MARL-eGCN [20]:** This method applies actor-critic framework to set tolls. We regard our roadnet as one zone in the method. It takes bounded action according to the state denoting the number of vehicles, and defines the rewards as the number of vehicles arriving at destinations.

5.4 Evaluation Metrics

- Throughput: The throughput calculates the number of vehicles that have arrived at the destination during the entire time period.
- Average Travel Time (ATT): For each vehicle, travel time measures the time between it entering and leaving the roadnet. ATT is the average travel time of all vehicles that have arrived at their destinations.
- Rewards: Global rewards over time, which is the sum of the global rewards of all time steps in Eq. 6.

5.5 Overall Performance (RQ1)

We conduct experiments on three real-world datasets, Hangzhou, Manhattan and Porto. The overall results are shown in Table 2. It is easy to observe that the proposed method CTRL outperforms other methods on three datasets (except that on Porto dataset, Δ-tolling method achieves very similar throughput as CTRL).

Generally, as expected, compared to the No-change method without tolling, traffic tolling is proven to be effective for traffic congestion. In addition, the dynamic tolling group of methods outperform the traditional Formula or Random method.

Among all the methods, Δ-tolling yields quite competing results in terms of throughput, while CTRL beat Δ-tolling with large margin on the other two measurements, ATT and Rewards. For the other methods, CTRL outperforms with large margin on all measurements.

5.6 Performance Gain with State Attention (RQ2)

To verify the effect of the proposed state attention mechanism in Section 4.3, we perform the ablation study comparing CTRL and CTRL without state attention. We choose the largest map (Porto) for the experiment. As shown in Fig. 6, CTRL achieves better performance in throughput and average travel time, compared with the version removing state-attention (using average to aggregate the state instead). The reason is that the state attention module can learn the appropriate weight on road states according to real-time traffic.

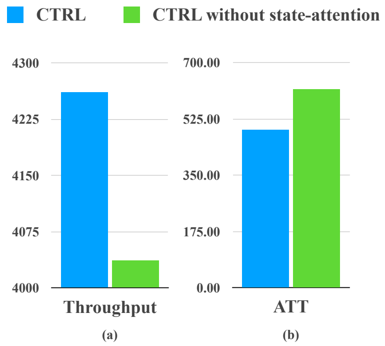

Figure 6: Effect of adding state-attention. Without the state-attention module (green), the total throughput of the road network is sharply reduced (a) and the average travel time of vehicles is significantly increased (b).

5.7 Tolling Fairness Induced by Q-Attention (RQ3)

To demonstrate the impact of the Q-Attention mechanism proposed in Section 4.4, we compare the prices given by CTRL and CTRL without Q-Attention at all time steps in Fig. 8. We find that the prices of the alternative routes under the same OD pair given by CTRL are clearly differentiated and not extremely high or low. However, without Q-Attention, the given prices (figure below) are too extreme and not fully distinguishable between certain route candidates of an OD pair (e.g., 1-0, 1-1). In practical applications, we expect prices to be relatively fair yet sufficiently distinguishable between routes, rather than too high or low.

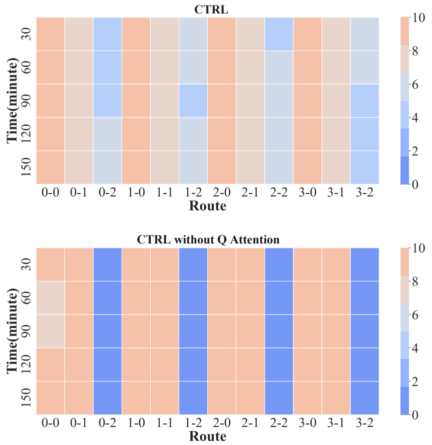

Figure 8: Comparison of the price distribution of CTRL and CTRL without Q-attention. The x-axis is "OD_id-route_id", the y-axis is time, and the values of heat map represent the corresponding price. CTRL is shown above.

5.8 Case Study (RQ4)

We show the performance of CTRL and baseline methods in terms of the number of vehicles and average speed on each route to illustrate why our method can admit more vehicles as shown in Table 2 in Porto. Note that, due to the space limitations, we only compare

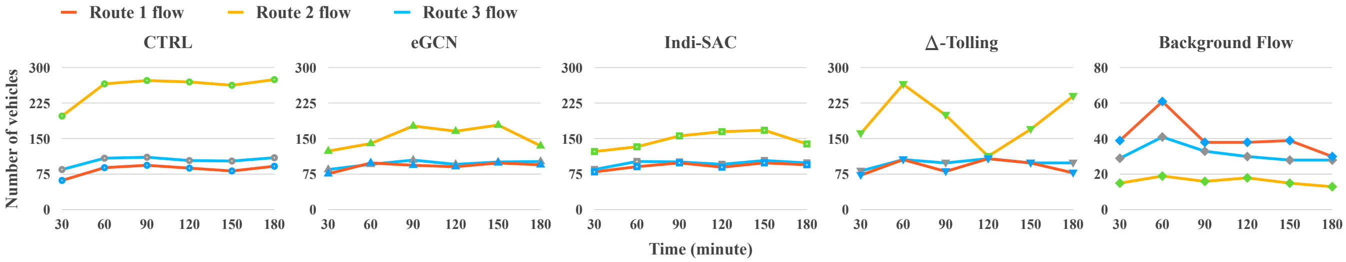

Figure 7: The routes of an OD over time steps in Porto. The points in different colors represent the number of vehicles on different routes over time. Each figure corresponds to a method, and the final figure is background flow that does not change route. As expected, the less background flow there is on the route, the more controlled flow chooses that route. Compared to baseline methods, CTRL makes greater use of capacity of the routes, as shown in Route 2.

CTRL with the other methods eGCN, Indi-SAC and Δ-tolling, which perform better among the baselines in Table 2.

Fig. 7 illustrates how CTRL assigns an OD's vehicles to different routes to accommodate more vehicles. We can observe that CTRL can adapt the route choice according to the change in background flow while other methods fail to do that. Although the route admits more vehicles, the route's travel efficiency, i.e., average vehicle speed shown in Fig. 9 is not affected.

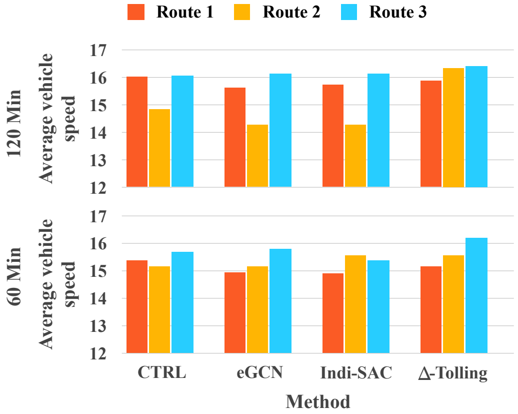

Figure 9: The average speed of vehicles on three routes at two time steps corresponds to Fig. 7. The average speed (maximum speed is $16.7m/s$) on the routes in CTRL is comparable to other methods, showing that there is no congestion on Route 2 caused by CTRL, which does make the best use of the capacity of the routes.

Further, we compare the total number of vehicles on different routes as in Fig. 10. CTRL has made great use of route 2 and finally achieves a much higher throughput than other methods.

Moreover, from the global view, we compare the total number of vehicles on this OD pair (all three different routes) in this scenario. We can observe that, among different time steps, CTRL can admit more vehicles than the baselines and hence support higher transportation efficiency.

6 CONCLUSION

In this paper, we formulate the dynamic tolling problem as a reinforcement learning problem and propose a method named CTRL to solve the problem. We innovate an interpretable reward design and an RL model with route-level state attention and competition-aware Q-attention to tackle the challenges in state representation and credit assignment. Compared with previous methods, CTRL can effectively maximize the throughput and minimize the average travel time on various datasets.

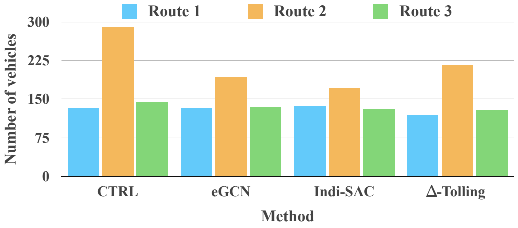

Figure 10: The number of vehicles on three routes of an OD in one time step. As expected, CTRL exploits the difference between route capacities at any time step to ensure a higher total number of vehicles of an OD, as shown in Fig. 11.

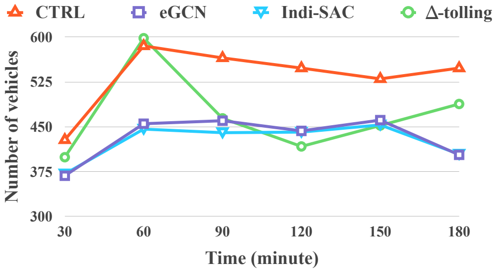

Figure 11: The number of vehicles in an OD pair over time step. CTRL admits more vehicles in almost all time steps, and cumulatively, compared with other baseline methods.

ACKNOWLEDGMENTS

This work was sponsored by Shanghai Pujiang Program under Grant No. 20PJ1409400, National Natural Science Foundation of China under Grant No. 62102246 and Provincial Key Research and Development Program of Zhejiang under Grant No. 2021C01034. This work was done in collaboration with Hangzhou Yunqi Academy of Engineering.

REFERENCES

[1] n.d.. OpenStreetMap.

[2] Mauricio Arango. 2019. Toll Road with Dynamic Congestion Pricing Using Reinforcement Learning. (2019).

[3] Kim Thien Bui, Vu Anh Huynh, and Emilio Frazzoli. 2012. Dynamic traffic congestion pricing mechanism with user-centric considerations. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 147–154.

[4] Yu-Han Chang, Tracey Ho, and Leslie P Kaelbling. 2004. All learning is local: Multi-agent learning in global reward games. (2004).

[5] Haipeng Chen, Bo An, Guni Sharon, Josiah Hanna, Peter Stone, Chunyan Miao, and Yeng Soh. 2018. Dyetc: Dynamic electronic toll collection for traffic congestion alleviation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[6] André de Palma, Moez Kilani, and Robin Lindsey. 2005. Congestion pricing on a road network: A study using the dynamic equilibrium simulator METROPOLIS. *Transportation Research Part A-policy and Practice* 39 (2005), 588–611.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[8] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).

[9] Ammar Haydari and Yasin Yilmaz. 2020. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[10] Junchen Jin and Xiaoliang Ma. 2019. A multi-objective agent-based control approach with application in intelligent traffic signal system. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3900–3912.

[11] Jiahui Jin, Xiaoxuan Zhu, Biwei Wu, Jinghui Zhang, and Yuxiang Wang. 2021. A dynamic and deadline-oriented road pricing mechanism for urban traffic management. *Tsinghua Science and Technology* 27, 1 (2021), 91–102.

[12] Dusica Joksimovic, Michiel CJ Bliemer, and Piet HL Bovy. 2005. Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. *Transportation Research Record* 1923, 1 (2005), 61–72.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[14] Zhenhui Li. n.d.. Reinforcement Learning for Traffic Signal Control. https://traffic-signal-control.github.io

[15] Jiancheng Long, Ziyou Gao, Haozhi Zhang, and Wai Yuen Szeto. 2010. A turning restriction design problem in urban road networks. *European Journal of Operational Research* 206, 3 (2010), 569–578.

[16] David Metz. 2018. Tackling urban traffic congestion: The experience of London, Stockholm and Singapore. *Case Studies on Transport Policy* 6, 4 (2018), 494–498.

[17] Hamid Mirzaei, Guni Sharon, Stephen Boyles, Tony Givargis, and Peter Stone. 2018. Enhanced delta-tolling: Traffic optimization via policy gradient reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 47–52.

[18] Venktesh Pandey and Stephen D Boyles. 2018. Multiagent reinforcement learning algorithm for distributed dynamic pricing of managed lanes. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2346–2351.

[19] Venktesh Pandey, Evana Wang, and Stephen D Boyles. 2020. Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. *Transportation Research Part C: Emerging Technologies* 119 (2020), 102715.

[20] Wei Qiu, Haipeng Chen, and Bo An. 2019. Dynamic Electronic Toll Collection via Multi-Agent Deep Reinforcement Learning with Edge-Based Graph Convolutional Networks.. In *IJCAI*. 4568–4574.

[21] Sandeep Saharan, Seema Bawa, and Neeraj Kumar. 2020. Dynamic pricing techniques for Intelligent Transportation System in smart cities: A systematic review. *Computer Communications* 150 (2020), 603–625.

[22] Guni Sharon, Michael W Levin, Josiah P Hanna, Tarun Rambha, Stephen D Boyles, and Peter Stone. 2017. Network-wide adaptive tolling for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies* 84 (2017), 142–157.

[23] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296* (2017).

[24] Richard Stuart Sutton. 1984. *Temporal credit assignment in reinforcement learning*. Ph. D. Dissertation. University of Massachusetts Amherst.

[25] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PloS one* 12, 4 (2017), e0172395.

[26] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1913–1922.

[27] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117* (2019).

[28] Hai Yang and Xiaoning Zhang. 2003. Optimal toll design in second-best link-based congestion pricing. *Transportation Research Record* 1857, 1 (2003), 85–92.

[29] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. *The World Wide Web Conference* (May 2019). https://doi.org/10.1145/3308558.3314139

[30] Xiaoning Zhang, H. Michael Zhang, Haijun Huang, Lijun Sun, and Tie-Qiao Tang. 2011. Competitive, cooperative and Stackelberg congestion pricing for multiple regions in transportation networks. *Transportmetrica* 7 (2011), 297 – 320.

[31] Pengjun Zhao and Haoyu Hu. 2019. Geographical patterns of traffic congestion in growing megacities: Big data analytics from Beijing. *Cities* 92 (2019), 164–174.

[32] Bojian Zhou, Michiel Bliemer, Hai Yang, and Jie He. 2015. A trial-and-error congestion pricing scheme for networks with elastic demand and link capacity constraints. *Transportation Research Part B: Methodological* 72 (2015), 77–92.