Model Free Perimeter Metering Control for Urban Networks Using Deep Reinforcement Learning

Dongqin Zhou
Department of Civil and Environmental Engineering
The Pennsylvania State University, University Park, PA, 16802
Email: dongqin.zhou@psu.edu

Vikash V. Gayah*
Department of Civil and Environmental Engineering
The Pennsylvania State University, University Park, PA, 16802
Email: gayah@engr.psu.edu
* Corresponding author.

Word Count: 7170 words + 1 table (250 words per table) = 7420 words

Submitted [*07/09/2020*]

ABSTRACT

Recent advances in regional traffic dynamics modeling have led to the development of large-scale traffic network control strategies, such as perimeter metering. However, existing perimeter control frameworks require precise knowledge of the functional forms of network Macroscopic Fundamental Diagrams (MFDs) or estimates of critical accumulation associated with regional congestion to be applied. In this paper, a completely model and data free deep reinforcement learning (DRL) based control scheme is proposed to tackle the optimal perimeter control problem for two-region urban networks governed by MFDs. Results of numerical tests show that the proposed method can learn optimal control policies in an extremely stable manner under various levels of uncertainties in the environment. The performance of the proposed scheme approaches that of the MPC in situations that the former is trained on. Moreover, the proposed method often exhibits superior performance to the MPC when deployed on unseen environments with different initial accumulations, traffic demands, and/or MFD modeling errors. The results in this paper suggest that DRL is a promising method for MFD-based network control.

Keywords: Macroscopic Fundamental Diagram (MFD); perimeter control; model free deep reinforcement learning (DRL)

INTRODUCTION

Urban traffic control is a difficult problem due to the underlying traffic dynamics. Modeling urban traffic network using microscopic modeling approaches is particularly difficult due to the high computational demands and complexity of urban traffic. Researchers have recently modeled urban traffic dynamics at an aggregate level using network Macroscopic Fundamental Diagrams (MFDs), which relate network productivity (e.g., average flow or rate trips can be completed) with network use (e.g., average density or accumulation of vehicles within a region). While network-wide relationships such as the MFD have been studied for quite some time (*1–4*), only recently have such relationships been integrated into a framework that could be used to model traffic dynamics evolution over time (*5*). The first empirical evidence that verified the existence of a network's MFD was provided in (*6*). Since then, extensive investigations have been performed regarding the existence and properties of MFD (Buisson and Ladier, 2009; Daganzo et al., 2011; Ji et al., 2010; Mazloumian et al., 2010a and others).

MFD-based frameworks for single or multi-region systems have been utilized to develop regional-level urban traffic control strategies, such as perimeter control. Perimeter control or cordon metering constrains the portion of flow allowed to transfer between two neighboring regions to improve overall throughput. (*5*) proposed a bang-bang perimeter control policy for single-region networks that aims to ensure the network never gets congested. While elegant, the success of this control policy is heavily dependent on the accuracy of the MFD model. Potential mismatch between the MFD model and environment dynamics was not considered. In light of this, a proportional-integral (PI) feedback regulator was developed in (*11*, *12*) for single-region perimeter control problems with and without time delays. The single-region perimeter control problem was also investigated using classic feedback control methods in multiple studies, such as (*13*) and (*14*). The optimal perimeter control problem for two-region urban networks was first formulated in (*15*). Since then, several studies have proposed extensions of multi-region perimeter control frameworks (*16–20*). However, solving the perimeter control problem in multi-region networks is a challenging task due to the problem complexity.

One promising method to solve perimeter control problems that has been shown to achieve state-of-the-art performances even with different levels of uncertainty in the environment is model predictive control (MPC) (*16*, *17*, *21–23*). This approach assumes that the MPC controller has sufficient knowledge to model traffic network dynamics. While errors can be accommodated, it still requires that the general functional form and scale of the regional MFDs to be known with a high level of accuracy. While several studies have proposed methods to estimate a network's MFD (*24–30*), such information is rarely available, as evidenced by the relatively small number of networks with empirically derived MFDs in the research literature. The MPC framework also may not adapt well to new environments since it assumes a horizon for predicting traffic dynamics (prediction horizon) and for estimating future control decisions (control horizon). These two parameters have to be determined beforehand, and it is unclear whether a particular parameter setting can transfer well in a new environment.

Other methods to solve the optimal perimeter control problem include linear quadratic regulator (*18*, *31*), multiple concentric PI controller (*19*), adaptive perimeter control (*32*, *33*), and others. However, these methods also assume full information of regional traffic dynamics and are prone to modeling errors. For this reason, a model free adaptive iterative learning perimeter control (MFAILPC) scheme was proposed in (*20*) that solved the perimeter control problem in a data-driven manner. However, information from the MFD—specifically, the critical accumulation—is still required in the controller design. Though the MFD and critical accumulation can be estimated from historical data, the estimations are likely to be inaccurate due to multivaluedness, instability, and hysteresis phenomena (*9*, *10*, *34*, *35*). In addition, the transferability of this method was not tested since in each case study the controller learns from scratch. It is therefore highly desirable to develop a method for perimeter control that is not only model free, but data free as well.

Reinforcement learning (RL) might be an appropriate technique to solve the perimeter control problem with less detailed knowledge on regional traffic dynamics. RL and DRL have recently been applied by the transportation community for a variety of traffic control purposes, most notably signalized

intersection control (*36–39*). An initial attempt to integrate RL into the solution for perimeter control problems can be found in (*40*). However, the RL method adopted in this paper is model-based and was only utilized to substitute the direct sequential method used in (*21*) after formulating the open-loop control problem into a nonlinear program. A separate study (*41*) also applied RL to perimeter control problems. However, this study takes metering rates obtained from an MPC-based framework as inputs and the RL agent was only used to redistribute these rates spatially around the cordon perimeters.

In this study, we propose a model free deep reinforcement learning perimeter control (MFDRLPC) scheme to solve the perimeter metering problem for a network made up of two regions. The proposed method learns the long-run impacts of specific gating decisions for every state that might arise, i.e., action value function or Q-value. Through an exploration process, different actions are tried to learn their impacts on the environment. This information is then provided back to the MFDRLPC agent so that it can improve upon its future decision-making. The process is completely model-free and requires no information that might otherwise be needed, such as the functional form of the MFD or impacts of vehicle routing decisions. Moreover, the proposed MFDRLPC is also data free and the DRL agents learn completely from interactions with the environment.

PROBLEM FORMULATION

This paper considers a heterogeneous network that can be partitioned into two regions, R_1 and R_2, that simulate the periphery of a city and the city center; see
Figure 1. Each region is assumed to be homogeneous with a well-defined MFD. Traffic demand with origin in R_i and destination in R_j at time t is denoted as $q_{ij}(t), i, j = 1,2$. Note that estimates of traffic demands are assumed to be known beforehand but actual demands can differ from these. The impact of errors between the actual and assumed demands will be explicitly tested within this framework. Denote as $n_{ij}(t)$ the accumulation in R_i with destination to R_j at time t. It follows $n_i(t) = \sum_j n_{ij}(t)$, where $n_i(t)$ is the total accumulation in R_i at time t.

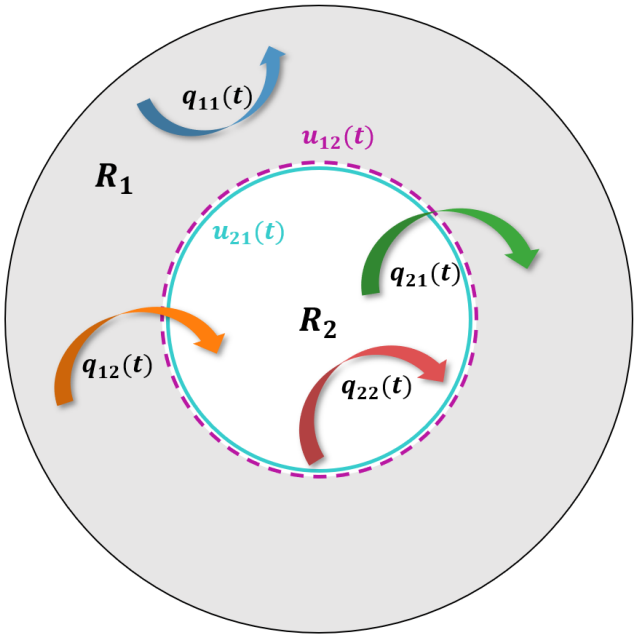

Figure 1. Two region MFDs system.

The MFD for R_i, denoted by $f_i(n_i(t))$, defines the trip completion rate within R_i as a function of its accumulation, $n_i(t)$. This MFD is then used to determine the trip completion rate of vehicles in R_i with

final destinations in R_j, denoted as $M_{ij}(t)$. Assuming that the average trip length is the same for all trips within a region, then $M_{ij}(t) = n_{ij}(t) \,/\, n_i(t) \cdot f_i\big(n_i(t)\big)$. Further, M_{ii} denotes the flow that vehicles reach their destinations, i.e., the flow that exits the network.

Perimeter controllers for the two-region system are assumed to exist on the border between the two regions with the goal of maximizing the number of vehicles that reach their destinations by any time t. The controllers, denoted by $u_{12}(t)$ and $u_{21}(t)$, where $u_{\min} \le u_{12}(t) \le u_{\max}$ and $u_{\min} \le u_{21}(t) \le u_{\max}$ with $0 \le u_{\min} < u_{\max} \le 1$, control the ratio of flow allowed to transfer from R_1 to R_2 and from R_2 to R_1 at time t, respectively.

Using this terminology, the two region perimeter control problem with MFDs is formulated as follows (similar to (*42*)):

$$J = \max_{u_{12}(t),u_{21}(t)} \int_{t_0}^{t_f} [M_{11}(t) + M_{22}(t)]dt \tag{1}$$

subject to:

$$\frac{dn_{11}(t)}{dt} = q_{11}(t) + u_{21}(t) * M_{21}(t) - M_{11}(t) \tag{2}$$

$$\frac{dn_{12}(t)}{dt} = q_{12}(t) - u_{12}(t) * M_{12}(t) \tag{3}$$

$$\frac{dn_{21}(t)}{dt} = q_{21}(t) - u_{21}(t) * M_{21}(t) \tag{4}$$

$$\frac{dn_{22}(t)}{dt} = q_{22}(t) + u_{12}(t) * M_{12}(t) - M_{22}(t) \tag{5}$$

$$M_{ij}(t) = \frac{n_{ij}(t)}{n_i(t)} f_i\big(n_i(t)\big) \tag{6}$$

$$n_{ij}(t) \ge 0, i, j = 1,2 \tag{7}$$

$$0 \le n_{11}(t) + n_{12}(t) \le n_{1,jam} \tag{8}$$

$$0 \le n_{21}(t) + n_{22}(t) \le n_{2,jam} \tag{9}$$

$$u_{min} \le u_{12}(t) \le u_{max} \tag{10}$$

$$u_{min} \le u_{21}(t) \le u_{max} \tag{11}$$

$$n_{ij}(t_0) = n_{ij,0}, i, j = 1,2 \tag{12}$$

where:

t_0: start time

t_f: final time

$n_{ij,0}$: initial accumulations at t_0

$n_{1,jam}$, $n_{2,jam}$: jam accumulation for R_1 and R_2

u_{min}, u_{max}: lower and upper bounds for $u_{12}(t)$ and $u_{21}(t)$

Equation (1) provides the objective function, i.e., to maximize the cumulative sum of vehicles reaching their destination and exiting the network at any time t. Note that doing so should also simultaneously minimize travel time within this network. Equations (2)-(6) describe traffic dynamics within the two-region system. Equations (7)-(9) provide minimum/maximum accumulations constraints within each region. Equations (10)-(11) define minimum/maximum control values, while (12) provides the initial accumulations. Note that the dynamics equations are only used for the simulation environment and not needed for the controller design.

METHODOLOGY

In this section, we first present the reinforcement learning (RL) environment where the perimeter control problem is reformulated. Then, the proposed MFDRLPC is explained, followed by the simulation environment that the DRL agent interacts with.

RL Formulation

The perimeter control problem can be formulated as a Markov decision process characterized by a tuple $< \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\pi}, \boldsymbol{\gamma} >$.

- **State space, $\boldsymbol{\mathcal{S}}$**. In this work, the state consists of the set of four accumulations n_{ij}, an estimate of the average demand q_{ij} during the next time step, and the implemented control values at the previous time step. To make scales of accumulation and demand consistent with implemented controller values, their quantities are divided by their respective maximum value.
- **Action space**, $\boldsymbol{\mathcal{A}}$. Three actions are defined for each of the two perimeter controllers: 1) increase its value by some amount, Δu; 2) keep the value unchanged; or, 3) decrease the value by Δu, where Δu is a predefined allowable change in controller values. Changing controller values by a set amount allows for a gradual change in control over time. In total, the agent has 9 actions to choose from (three options for each of the two controllers). After an action is chosen, it stays effective in the environment for the duration of a time step, Δt.
- **State transition function**, $\boldsymbol{\mathcal{P}}$. Given the observed state S_t and chosen action a_t, the system arrives at a new state S_{t+1}, according to the state transition function $\mathcal{P}(S_{t+1}|S_t, a_t)$: $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$.
- **Reward function**, $\boldsymbol{\mathcal{R}}$. The agent receives an immediate reward from the environment at time step $t+1$ after taking an action at time step t, according to a reward function $\mathcal{R}(S_t, a_t)$: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The trip completion rate in time step t is $M_{11}(t) + M_{22}(t)$, which serves as the reward for the agent and higher rewards are preferred. However, the rewards are scaled to a value between 0 and 1 since previous literature suggests proper reward scaling provides more stable learning processes for DRL agents (*43*). The reward function is thus defined as $\frac{M_{11}(t) + M_{22}(t)}{C}$, where C is a large constant. Moreover, a large negative penalty is added to the reward if the actions chosen by the MFDRLPC agents lead to gridlock or invalid accumulation values in the two-region system.
- **Policy**, $\boldsymbol{\pi}$, and **discount factor**, $\boldsymbol{\gamma}$. At each time step t, the agent chooses an action based on a policy parameterized by θ, π_θ: $\mathcal{S} \to \mathcal{A}$, with the purpose of maximizing the expected return $\mathbb{E}[G_t]$. The return G_t is defined as the total discounted reward from time step t

$$G_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} \mathcal{R}_{\tau+1} \tag{13}$$

where T is the total time steps of an episode and $\gamma \in [0,1]$ scales down the importance of future rewards. Intuitively, future rewards involve increasing uncertainty and are valued less than immediate rewards.

Algorithm

In model free RL methods, the action value function $Q(S, a)$, also known as Q-value, is used to approximate the expected return. Specifically, the action value function is defined as the expected return starting from state s, taking action a, and then following the current policy. Once $Q(S, a)$ is known, an optimal control policy can be derived by taking greedy actions that maximize the Q-value at any given state, i.e.,

$$\pi(S_t) = \arg\max_{a_t} Q(S_t, a_t) \tag{14}$$

Tabular methods such as Q-learning (*44*) and SARSA (*45*) store Q-values in a table where they can be iteratively updated according to the Bellman Equation (*46*). However, these methods cannot scale well with large state spaces since it is intractable to enumerate all possible state-action pairs.

With the active progress made in the deep learning community, researchers have proposed to use neural networks to estimate the action value function. Nevertheless, RL is known to be extremely unstable and sometimes diverge when nonlinear functions are used to approximate action value function (i.e., the deadly triad issue (*46*)). Deep Q-learning (*47*) is the first work that successfully addressed this issue and achieved stability through the use of experience replay and target network, where the seminal Deep Q-Network algorithm (DQN) was proposed. The learning target of deep Q-learning is

$$Y_t = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t^-) \quad (15)$$

where $Q(:,:,\boldsymbol{\theta}_t^-)$ is the target network parameterized by $\boldsymbol{\theta}_t^-$. The target network has the same structure as the Q-network, but its weights are only updated periodically with Q-network parameters.

As can been observed in (15), updates of action values in deep Q-learning include a maximization operation, which results in an overestimation of action values. This overestimation problem was first observed in (*48*) and later affirmatively studied in (*49*). In the latter reference, a new algorithm named Double DQN was developed that effectively addressed the overestimation issue by combining Double Q-learning (*50*) with DQN. Specifically, the max operation in the learning target (15) is decomposed into action selection and evaluation. The Q-network is used for action selection and the target network for evaluation. In mathematical term, the learning target is

$$Y_t = R_{t+1} + \gamma Q(S_{t+1}, \arg\max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t^-) \quad (16)$$

where $Q(:,:,\boldsymbol{\theta}_t)$ represents the Q-network parametrized by $\boldsymbol{\theta}_t$. In this work, Double DQN is adopted as the learning algorithm.

DRL agents learn purely from interactions with the environment and a significant amount of experiences are needed. A distributed reinforcement learning architecture named Ape-X was proposed in (*51*) that maintains numerous actors and a single centralized learner. Each actor has its own instance of the environment. The actors are assigned different exploration strategies to expand the amount of experiences they jointly encounter. The experiences are then stored in a shared replay buffer. The learner samples experiences from the buffer in a prioritized fashion (*52*) and updates the Q-network. The actors are then updated with the most up-to-date parameters from the Q-network. As the actors gather more and more experiences by interacting with the environment, the learner updates the Q-network to generate increasingly accurate estimates of the true action value function $Q(S, a)$. When the action value function has been fully learnt, an optimal policy can then be derived according to (14). Interested readers are referred to (*51*) for more information about the Ape-X structure.

In this paper, the Ape-X architecture is combined with Double DQN. Instead of prioritizing experiences with TD errors (*51*, *52*), we prioritize based on the recency of experiences, i.e., the experiences where the DRL agents are making more educated decisions are valued more than outdated experiences. When the amount of gathered experiences exceeds the replay buffer size, the old experiences are removed from the replay buffer. Further, all actors use decaying ϵ-greedy policies for exploration. Pseudocode for the proposed model free deep reinforcement learning perimeter controller is presented in Algorithm 1.

Simulation Environment

In the context of DRL, it is requisite to have an environment which the agent could interact with to learn the expected rewards for various actions taken at each state. The agent internalizes the environment's dynamics via this interaction. The agent also receives sequential rewards from the environment, which determine the agent's behavior. In this paper, the simulation environment is expressed by the MFDs plant as described in (*21*). As this reference pointed out, the traffic dynamics in the MFDs plant are different from that in the MFDs prediction model. In the MFDs plant, noises in the demand and errors in the MFDs are expected, which represent real-world issues. In this paper unbiased noises in demand and errors in MFDs are considered.

Noises in demand are modeled as follows:

$$\tilde{q}_{ij}(t) = \max\left(q_{ij}(t) * \left(1 + \varepsilon_{ij}(t)\right), 0\right), i, j = 1,2 \quad (17)$$

where $\tilde{q}_{ij}(t)$ is actual demand in the environment, $q_{ij}(t)$ is the average demand provided to the MFDRLPC agents (or MPC) and $\varepsilon_{ij}(t)$ is a Gaussian error term with mean 0 and standard deviation σ. In this way, the error in demand is a percentage of the average demand that simulates a temporal fluctuation of the actual demand. For the same value of σ, the magnitude of potential errors in demand would increase with the expected demand level.

Algorithm 1: Model Free Deep Reinforcement Learning Perimeter Control (MFDRLPC)

Initialize Q-network $\boldsymbol{\theta}_0$, replay buffer **B**, memory size M, iteration number I
for *iter* = 1 **to** I **do**
  for all actors **do**
    Load Q-network $\boldsymbol{\theta}_{iter} = \boldsymbol{\theta}_{iter-1}$
    $S_0 \leftarrow$ Environment.Reset()
    for $t = 1$ **to** T **do**
      $a_{t-1} = \pi_{\theta_{iter}}(S_{t-1})$
      $(R_t, S_t) \leftarrow$ Environment.Step(a_{t-1})
      B.add($(S_{t-1}, a_{t-1}, R_t, S_t)$)
    end for
  end for
  if B.size() > M **then**
    B.remove()
  end if
  Training sample ← **B**.sample()
  Periodically load target network $\boldsymbol{\theta}^{-}_{iter} = \boldsymbol{\theta}_{iter-1}$
  $\boldsymbol{\theta}_{iter} \leftarrow$ Update Q-network towards learning target (16)
end for

Errors in the MFD are modeled as:

$$\tilde{f}_i\big(\tilde{n}_i(t)\big) = f_i\big(\tilde{n}_i(t)\big) + \varsigma_i * \tilde{n}_i(t), i = 1,2 \tag{18}$$

where $\tilde{n}_i(t)$ is the accumulation in R_i at time t, $\varsigma_i \sim U(-\alpha, \alpha)$ is an error term sampled from a uniform distribution with predefined error level α. This suggests that errors between the expected and realized trip completion rates grow as the network gets more congested, which is consistent with empirical findings and analytical studies (*7*, *10*).

Traffic dynamics in the environment are then computed as described in Equations (2)-(6), except MFDs and demand values in these equations are replaced with the terms (17)-(18). Solution to these dynamic equations yields accumulations at the next time step as well as number of trips completed, which are used to calculate rewards for the agent.

In summary, the simulation environment is a two-region MFDs plant expressing traffic dynamics, where there are noises in the demands and/or errors in the MFDs. The environment implements an action generated by the agent and arrives at a new state. It also returns rewards to the agent that evaluate the actions generated.

NUMERICAL TESTS

In this section, the proposed MFDRLPC is tested and compared with the MPC framework by solving the two-region optimal perimeter control problem. Note that the benefit of the proposed method is that information about the MFD or knowledge of system dynamics are not needed, whereas the MPC framework requires that the MFD and dynamics equations that govern the evaluation of network accumulations and trip completions to be fully known.

Experiment Setup

The MFD of Yokohama, Japan, is adopted from (*6*, *53*) for R_1:

$$f_1(n) = \begin{cases} 2.28 \times 10^{-8} n^3 - 8.62 \times 10^{-4} n^2 + 9.58n, & 0 \le n < 14{,}000 \\ 27{,}731 - 1.38655(n - 14000), & 14{,}000 \le n \le 34{,}000 \\ 0, n > 34{,}000 \end{cases} \tag{19}$$

where critical accumulation $n_{1,cr} = 8241$ veh, jam accumulation $n_{1,jam} = 34000$ veh, maximum trip completion rate $C_{1,max} = f_1(n_{1,cr}) = 33168$ veh/hr. For R_2, the above MFD is scaled down by a factor of 2 to simulate a smaller region so $n_{2,cr} = 4120$ veh, $n_{2,jam} = 17000$ veh, $C_{2,max} = f_2(n_{2,cr}) = 16584$ veh/hr.

The traffic demand pattern adopted in this paper is shown in Figure 2(a), which simulates a 1-hour morning peak where there is a larger demand to R_2 (city center) than to R_1 (the periphery of a city). The duration of a time step is set to $\Delta t = 60s$ that simulates the cycle length of traffic signals. The signals can be placed in the border between two regions to implement the perimeter controllers.

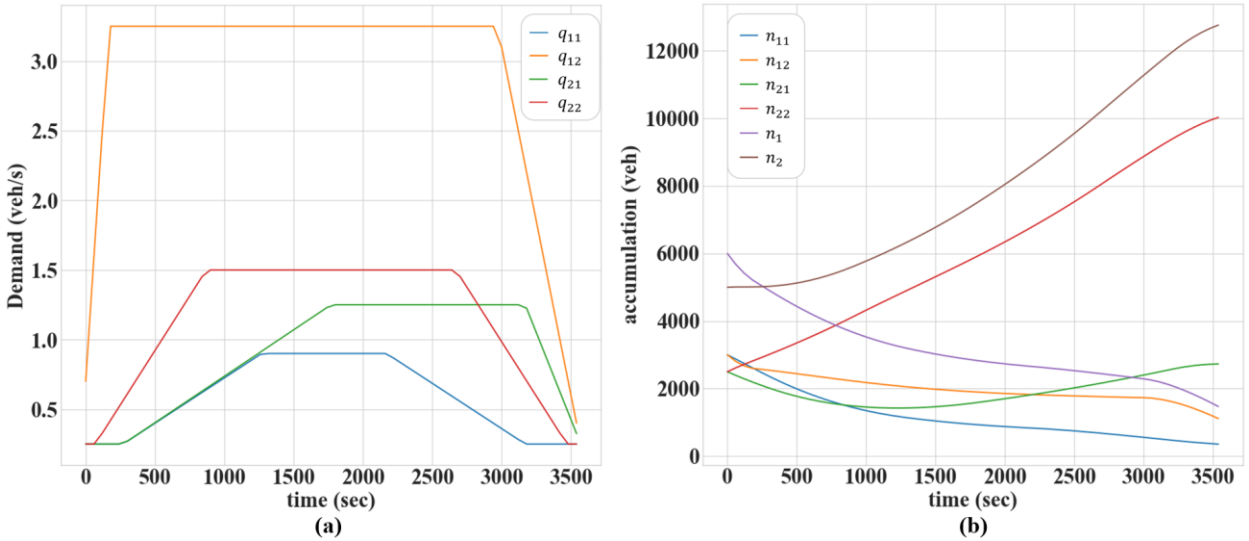

Figure 2. (a) Average demands used for numerical tests; (b) Accumulations that arise under no control

Initial accumulations are assumed to be $n_{11,0} = n_{12,0} = 3000$ veh and $n_{21,0} = n_{22,0} = 2500$ veh. Intuitively, when the initial accumulations are too small, no control should be applied, and the controller values will be set to the maximum value possible. On the contrary, no control can prevent gridlock when the initial accumulations are extremely large.

The evolution of accumulations when the demands and the MFDs are deterministic and no control is applied is presented in Figure 2(b). As can be observed, the accumulation in R_1 steadily decreases while the accumulation in R_2 keeps increasing and approaches jam accumulation. These results are expected since the demand from R_1 to R_2 is significantly larger than otherwise. Clearly, this is unsustainable and undesirable, even more so considering region 2 simulates the city center. In the next section it will be shown how this congestion can be effectively addressed by properly implementing perimeter control.

The objective of perimeter control is to maximize the number of trips completed. Based on this objective, two metrics are defined to evaluate the performance of the proposed MFDRLPC and MPC: a) total travel time (TTT) of all vehicles in the system, which can be calculated as the area between the arrival curve and departure curve in an input-output diagram; b) cumulative total trip completion (CTC) during the 1-hour period under study.

Experiment Results

Multiple MFDRLPC agents were trained in the experiments, each under different levels of errors that might exist both in the demands and in the MFDs, as described in Table 1. Note that, for all agents trained, the same Q-network design and same set of hyperparameters were used. The allowable change in perimeter controller values is set to $\Delta u = 0.1$, which is a reasonable control precision.

Table 1. MFDRLPC agent configurations

Agent No.	σ	α	Description
1	0	0	deterministic scenario as a benchmark
2	0.1	0	test the performance under medium noise in demands
3	0.2	0	test the performance under high noise in demands
4	0	0.2	test the performance with errors in MFDs
5	0.1	0.2	test the performance with mixed errors in demands and MFDs

Stability of the MFDRLPC method

Figure 3 shows the evolution of CTC achieved in the 1-hour period with training iterations for the five agents. The results are obtained by training the agents with 10 random seeds, while all hyperparameters are fixed. The darker line shows the median performance over random seeds. The shaded areas of MFDRLPC curves represent the confidence bound of the performances and are obtained by plotting the two extreme values over random seeds in each iteration. Moreover, the performances of MPC are also plotted in Figure 3. In cases where there are noises in demand and/or errors in MFDs, the MPC is run for 10 times and the median and extreme values are reported. Since the MPC is a model-based method that involves no learning, its range of performances is relatively fixed. As shown in Figure 3, the proposed MFDRLPC agents can learn perimeter control strategies under all training scenarios in an extremely stable fashion. The performances of MFDRLPC agents approach those of MPC most of the time and sometimes even exceed the performance of MPC when there is uncertainty from demands or MFDs. In a few training instances, the performances of MFDRLPC agents have not fully converged within 250 iterations (i.e., still improving). Thus, the performances may be even better if the agents are allowed to train for longer periods.

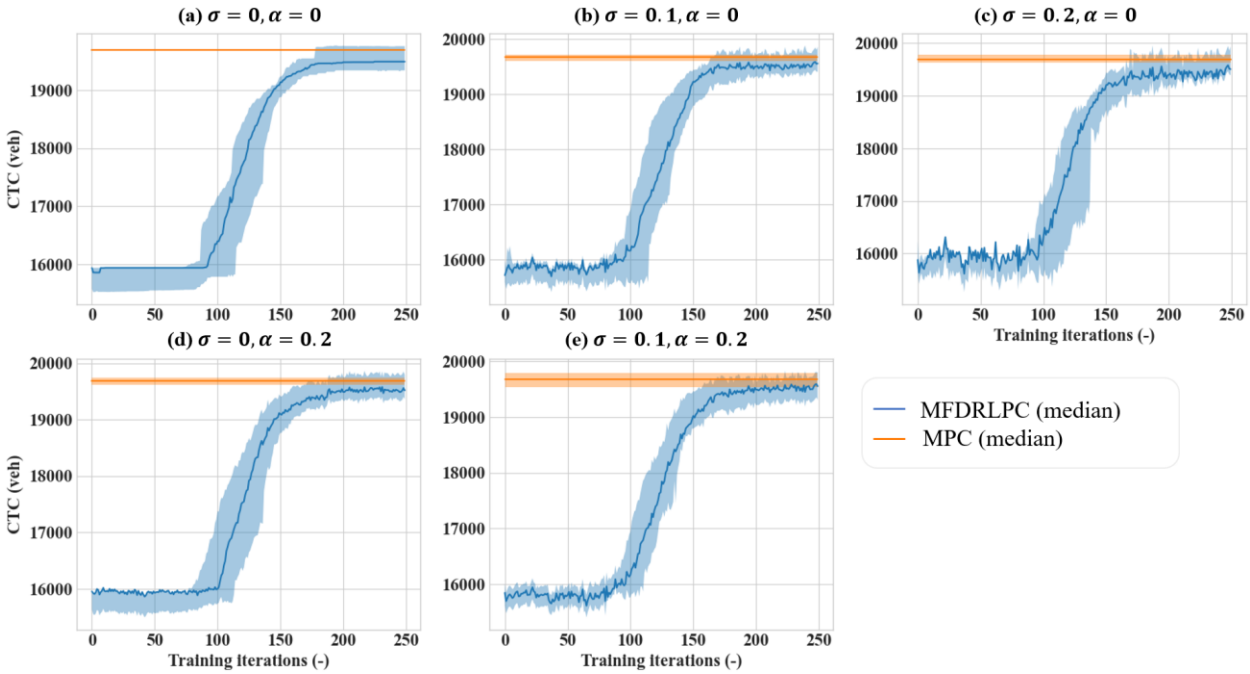

Figure 3. Learning curves of 5 MFDRLPC agents

Effectiveness of the MFDRLPC method

The proposed MFDRLPC is compared with the state-of-the-art MPC approach to examine its effectiveness. The MPC controller is implemented as per (*21*) without adding smoothing control constraints, and it assumes a prediction horizon of 20 and a control horizon of 2. The lower bound for the perimeter controller

values is set to $u_{\min} = 0.1$ and upper bound set to $u_{\max} = 0.9$. Additionally, the results when no control (NC) is applied are provided as a baseline. Figure 4 and Figure 5 show the evolution of accumulations and control actions over the study period when there is no uncertainty in the environment. Note that since the overall demands to R_1 are smaller and R_1 has larger capacity to contain vehicles, transfers from R_2 to R_1 are not restricted by any control method. Thus, $u_{21} = u_{\max}$ for the entire study period and this is not shown in Figure 5. For scenarios where uncertainties are present, the results are similar.

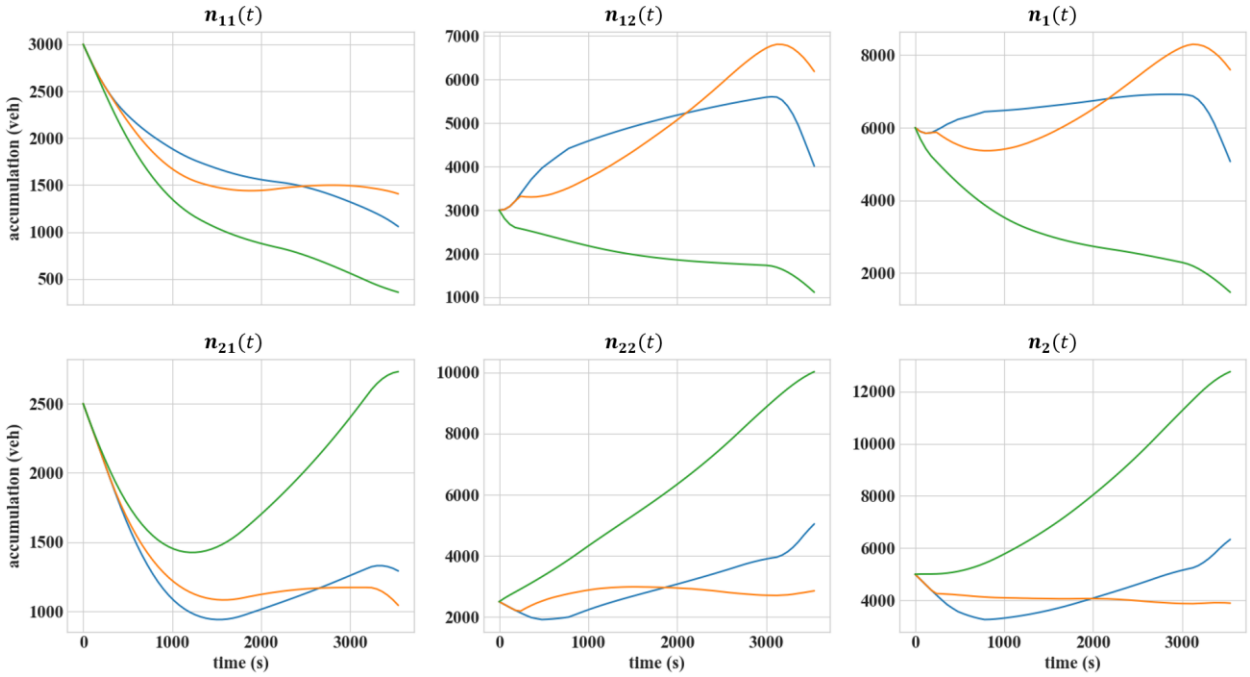

Figure 4. Evolution of accumulation under determinism. Blue: MFDRLPC; Orange: MPC; Green: NC

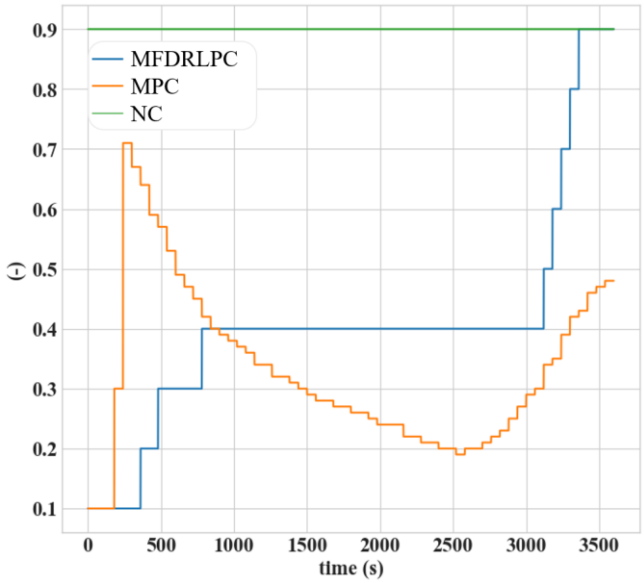

Figure 5. Control action (u_{12}) over time under determinism

When no control is applied, the accumulation in R_2 rapidly increases and approaches gridlock. This is mostly due to the high demands to R_2 both from R_1 and internally from R_2. On the contrary, the

accumulation in R_1 steadily decreases since vehicles in R_1 can pass the border freely and demands to R_1 are much lower. Under perimeter control, only a portion of vehicles from R_1 can transfer to R_2, hence n_{12} first increases. Then, when the demands decrease at the end of the simulated morning peak, more vehicles are allowed to transfer and n_{12} decreases. Due to the relatively stricter control actions chosen by MFDRLPC in the early peak, the accumulation in R_2 first decreases. Later, as the controller gets relaxed, more vehicles are able to enter R_2 and its accumulation increases. Overall, the evolutions of accumulations exhibit similar trends for both MPC and MFDRLPC.

Figure 5 reveals that both MPC and MFDRLPC tend to increase u_{12} to allow more transfer flows in the early stage when demands are relatively small. As demands increase to the maximum, transfer flows are more tightly restricted. Approaching the end of the morning peak, all controllers have a propensity to allow more vehicles to transfer and complete their trips. Furthermore, instead of decreasing u_{12} at higher demand and increasing it later as MPC controller does, MFDRLPC keeps u_{12} around the mean value of MPC control values. In this way, MFDRLPC is able to increase the number of trips completed as well as reduce the complexity of perimeter control implementation in real life.

Transferability of the MFDRLPC method

This section examines the transferability of the MFDRLPC to unseen scenarios. To the best of our knowledge, this is the first examination of the ability of perimeter control methods to be implemented on unseen environments. Under all case studies reported in (*20*), the MFAILPC controller learns from scratch and it is unclear whether the controller in one case can perform well in another without conducting the learning process all over again. MPC (*21*) is a model-based method and needs to formulate an optimization problem at every time step. It remains in question whether a particular parameter setting (control horizon, prediction horizon, etc.) can generalize well to a different environment.

A variety of environment configurations are considered, each differing in at least one of these three factors: initial accumulations, demand patterns, and MFD models. The transferability of the proposed method is tested against each factor by keeping the other two constant.

The initial accumulations in the environment are chosen to simulate daily variation of traffic conditions according to

$$n_{i,new} = n_{i,0} * (1 + \phi), i = 1,2 \quad (20)$$

where ϕ increases from 0 to 0.30 by 0.05 representing the variation of initial accumulations from original accumulations $n_{i,0}$. Note that $n_{1,0} = 6000$ veh and $n_{2,0} = 5000$ veh. The demand patterns are selected by

$$\tilde{q}_{ij} = q_{ij} * (1 + \eta), i, j = 1,2 \quad (21)$$

where q_{ij} is basic demand from Figure 2a and $\eta = 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30$, which indicates the variation of traffic demands. The MFD models in the environment are defined as

$$\tilde{f}_i(\tilde{n}_i(t)) = f_i(\tilde{n}_i(t)) + \varphi * \tilde{n}_i(t), i = 1,2 \quad (22)$$

where $\varphi \in [-0.50, -0.30, -0.10, 0, 0.10, 0.30, 0.50]$ and it indicates the level of MFD modeling errors. When φ is less than 0, the environment dynamics internalized by the MFDRLPC agents or known by the MPC overrepresent the real situation in the environment and fewer trips can be completed. In total, $7^3 = 343$ new environment configurations are considered, and only the results for a fraction of environment configurations are presented. Results for other configurations are similar and do not affect the conclusions.

The performances of the proposed method and the MPC are expressed as improvements from the NC in terms of total travel time (TTT), as shown in Figure 6. The environments assume $\eta = 0, \varphi = 0$. As can be observed from Figure6, the improvements from NC for all MFDRLPC agents and MPC increase with initial accumulations, which suggests a higher level of necessity to implement perimeter control as initial accumulations become larger. More importantly, the performances of the MFDRLPC agents are generally better than those of the MPC as initial accumulations increase. For example, MFDRLPC agents 1, 2, 3, and 5 consistently outperform MPC when $\phi \geq 0.15$ even though the agents have never encountered the test environments before and MPC has full information about the environment at every time step. Agent 1 was trained without environment uncertainty, i.e., $\sigma = 0, \alpha = 0$. The test environment adopts basic demand pattern and no MFD modeling error, which is consistent with the environment agent 1 was trained

on. The excellent performance achieved by agent 1 suggests that the MFDRLPC can adapt well to similar environments. Further, agents 2, 3, and 5, which were trained with high environment uncertainty, also performed very well on the test environments. This indicates that an MFDRLPC agent trained in high uncertainty can generalize well to environments with low uncertainty. Note that, though agent 4 underperforms other agents, it generally performs as well or better than MPC.

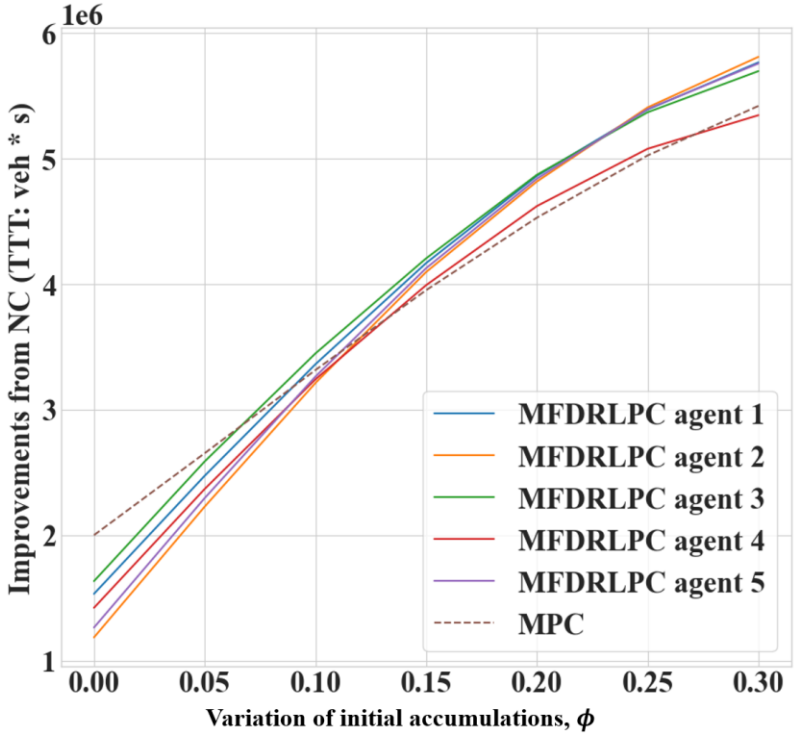

Figure 6. Performances with respect to deviation of initial accumulations

To test the sensitivity of the proposed MFDRLPC to unseen demands in the environment, initial accumulations are kept at their original values and MFDs are assumed to have no modeling error while the demand levels are allowed to vary. Figure 7 shows the improvements from NC with respect to demand for the five MFDRLPC agents and MPC. As shown in Figure 7, the improvements from NC monotonically increase with traffic demands for both the MFDRLPC and MPC. Initially, when traffic demands are small, MPC exhibits some advantage over MFDRLPC agents. However, as traffic demand becomes relatively large ($\eta \geq 0.15$), MPC cannot generalize as well and is outperformed by MFDRLPC agents. Notably, agent 3, which was trained assuming only a noise level of $\sigma = 0.2$ in the environment, consistently achieved the best performance when $\eta \geq 0.15$. This seems reasonable since its training environment is closest to the test environments that assume increasing demands and no MFD modeling errors. Additionally, agent 2, whose training environment ($\sigma = 0.1, \alpha = 0$) is second closest to the test environments, achieved second best performances as traffic demands increase. It is then shown again that the MFDRLPC agents can transfer well to similar environments. The relative consistency of performances among all agents also confirms the robustness of the proposed method to traffic demands in the environment.

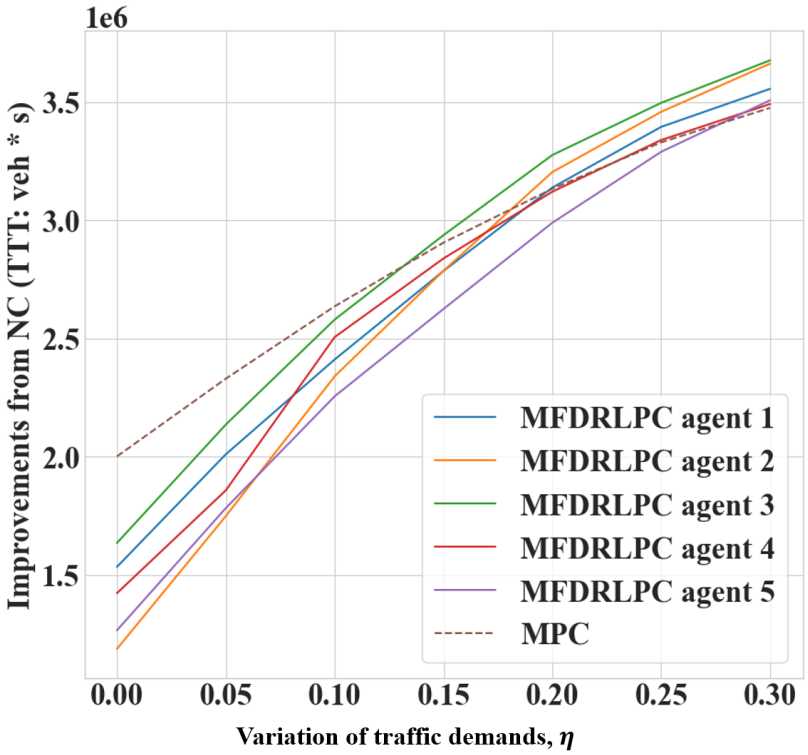

Figure 7. Performances with respect to traffic demands

The performances of all methods relative to NC with respect to MFD modeling errors are shown in Figure 8. The test environments take original initial accumulations and basic traffic demands. As MFD modeling error changes from -0.5 to 0.5, the trip completion rate increases, hence the number of trips completed in the 1-hour period increases and total travel time in the system decreases. Under most MFD modeling error levels, MPC performs the best. However, when the MFDs are significantly underestimated ($\varphi \geq 0.4$), improvements from NC turn negative for both MPC and the MFDRLPC agents, suggesting that it is best not to implement any perimeter control at all. Intuitively, when the MPC or MFDRLPC agents assumes that traffic dynamics will evolve according to the underestimated or under-perceived MFD, perimeter control will be implemented to keep regions from becoming congested. However, since the network is more productive than expected, better performances can be obtained by not restricting vehicle movement. Moreover, when the MFDRLPC agents are over-optimistic about the environment dynamics, i.e., environment MFDs are overestimated or $\varphi < 0$, the differences of performances between the MFDRLPC and MPC become smaller. On the other hand, the performances between the MFDRLPC and MPC tend to diverge as MFD underestimation error grows. This shows the proposed MFDRLPC is robust to MFD overestimation error in the environment.

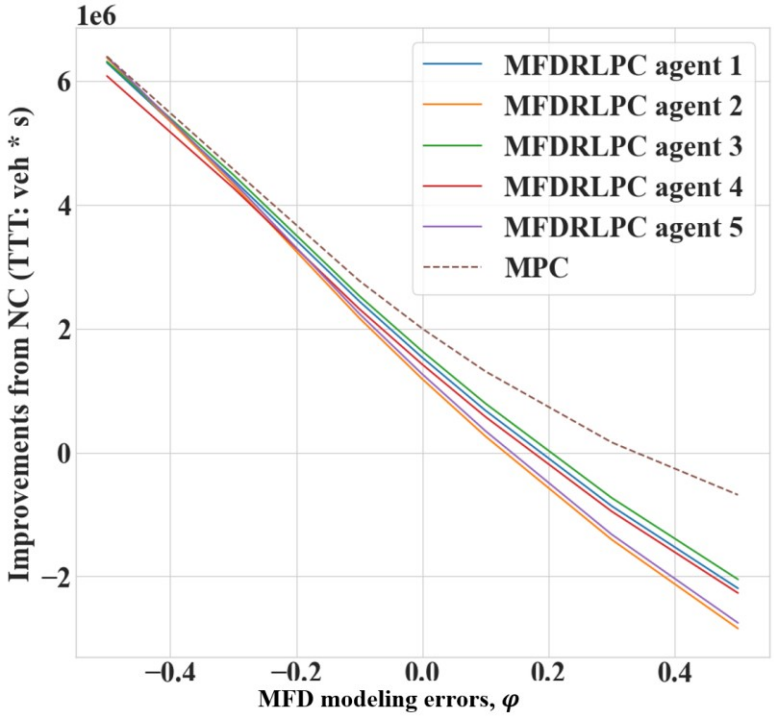

Figure 8. Performances with respect to MFD modeling errors

Overall, the test results presented in this section demonstrate the superior transferability of the proposed method and show that DRL is a promising method for MFD-based network control. Particularly, the proposed MFDRLPC is robust to different initial accumulations, traffic demands, and MFD overestimation errors in the environment. These results also suggest that the MFDRLPC agents be trained offline with relatively large initial accumulations, high traffic demands, and on environments with higher MFDs, so that they can be deployed directly to environments that have lower outflows while maintaining performances that are comparable with those of MPC.

SUMMARY

In this paper, a DRL-based method named MFDRLPC is developed to solve the optimal perimeter control problem for two-region urban networks with MFDs by combining the Ape-X architecture and Double DQN. The proposed method is completely model free and data free. Results from extensive numerical experiments show that the proposed MFDRLPC can learn in an extremely stable manner and achieve performances that are comparable with or better than the state-of-the-art MPC-based framework. Moreover, the proposed MFDRLPC is shown to be highly transferable and robust by deploying it to a variety of test environments with different initial accumulations, traffic demands, and MFD modeling errors. In addition, results reported in this paper provide a lower bound for the full capability of DRL-based methods on perimeter control. With future research efforts, even better results can be achieved. In general, this paper demonstrates that DRL has great applicability on MFD-based network traffic control. Future works could include the development of a general multi-region DRL-based control framework that solves perimeter control and traffic signal control simultaneously.

ACKNOWLEDGEMENTS

This research was supported by NSF Grant CMMI-1749200 and a seed grant through the Penn State Institute of CyberScience.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: VG, DZ; analysis and interpretation of results: VG, DZ; draft manuscript preparation: VG, DZ. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Godfrey, J. W. The Mechanism of a Road Network. *Traffic Engineering & Control*, Vol. 11, No. 7, 1969, pp. 323–327.
2. Smeed, R. J. The Road Capacity of City Centers. *Traffic Engineering & Control*, Vol. 9, No. 7, 1967, pp. 455–458.
3. Mahmassani, H., J. C. Williams, and R. Herman. Investigation of Network-Level Traffic Flow Relationships: Some Simulation Results. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 971, 1984, pp. 121–130.
4. Mahmassani, H., J. C. Williams, and R. Herman. Performance of Urban Traffic Networks. 1987.
5. Daganzo, C. F. Urban Gridlock: Macroscopic Modeling and Mitigation Approaches. *Transportation Research Part B: Methodological*, Vol. 41, No. 1, 2007, pp. 49–62. https://doi.org/10.1016/j.trb.2006.03.001.
6. Geroliminis, N., and C. F. Daganzo. Existence of Urban-Scale Macroscopic Fundamental Diagrams: Some Experimental Findings. *Transportation Research Part B: Methodological*, Vol. 42, No. 9, 2008, pp. 759–770.
7. Buisson, C., and C. Ladier. Exploring the Impact of Homogeneity of Traffic Measurements on the Existence of Macroscopic Fundamental Diagrams. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2124, No. 1, 2009, pp. 127–136. https://doi.org/10.3141/2124-12.
8. Ji, Y., W. Daamen, S. Hoogendoorn, S. Hoogendoorn-Lanser, and X. Qian. Investigating the Shape of the Macroscopic Fundamental Diagram Using Simulation Data. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2161, No. 1, 2010, pp. 40–48. https://doi.org/10.3141/2161-05.
9. Mazloumian, A., N. Geroliminis, and D. Helbing. The Spatial Variability of Vehicle Densities as Determinant of Urban Network Capacity. Vol. 368, No. 1928, 2010, pp. 4627–4647. https://doi.org/10.1098/rsta.2010.0099.
10. Daganzo, C. F., V. V. Gayah, and E. J. Gonzales. Macroscopic Relations of Urban Traffic Variables: Bifurcations, Multivaluedness and Instability. *Transportation Research Part B: Methodological*, Vol. 45, No. 1, 2011, pp. 278–288. https://doi.org/10.1016/j.trb.2010.06.006.
11. Keyvan-Ekbatani, M., A. Kouvelas, I. Papamichail, and M. Papageorgiou. Exploiting the Fundamental Diagram of Urban Networks for Feedback-Based Gating. *Transportation Research Part B: Methodological*, Vol. 46, No. 10, 2012, pp. 1393–1403. https://doi.org/10.1016/j.trb.2012.06.008.
12. Keyvan-Ekbatani, M., M. Papageorgiou, and V. L. Knoop. Controller Design for Gating Traffic Control in Presence of Time-Delay in Urban Road Networks. *Transportation Research Part C: Emerging Technologies*, Vol. 59, 2015, pp. 308–322. https://doi.org/10.1016/j.trc.2015.04.031.
13. Haddad, J., and A. Shraiber. Robust Perimeter Control Design for an Urban Region. *Transportation Research Part B: Methodological*, Vol. 68, 2014, pp. 315–332. https://doi.org/10.1016/j.trb.2014.06.010.
14. Haddad, J. Optimal Coupled and Decoupled Perimeter Control in One-Region Cities. *Control Engineering Practice*, Vol. 61, 2017, pp. 134–148. https://doi.org/10.1016/j.conengprac.2017.01.010.
15. Haddad, J., and N. Geroliminis. On the Stability of Traffic Perimeter Control in Two-Region Urban Cities. *Transportation Research Part B: Methodological*, Vol. 46, No. 9, 2012, pp. 1159–1176. https://doi.org/10.1016/j.trb.2012.04.004.

16. Hajiahmadi, M., J. Haddad, B. De Schutter, and N. Geroliminis. Optimal Hybrid Perimeter and Switching Plans Control for Urban Traffic Networks. *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 2, 2015, pp. 464–478. https://doi.org/10.1109/TCST.2014.2330997.
17. Haddad, J., M. Ramezani, and N. Geroliminis. Cooperative Traffic Control of a Mixed Network with Two Urban Regions and a Freeway. *Transportation Research Part B: Methodological*, Vol. 54, 2013, pp. 17–36. https://doi.org/10.1016/j.trb.2013.03.007.
18. Aboudolas, K., and N. Geroliminis. Perimeter and Boundary Flow Control in Multi-Reservoir Heterogeneous Networks. *Transportation Research Part B: Methodological*, Vol. 55, 2013, pp. 265–281. https://doi.org/10.1016/j.trb.2013.07.003.
19. Keyvan-Ekbatani, M., M. Yildirimoglu, N. Geroliminis, and M. Papageorgiou. Multiple Concentric Gating Traffic Control in Large-Scale Urban Networks. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 4, 2015, pp. 2141–2154. https://doi.org/10.1109/TITS.2015.2399303.
20. Ren, Y., Z. Hou, I. I. Sirmatel, and N. Geroliminis. Data Driven Model Free Adaptive Iterative Learning Perimeter Control for Large-Scale Urban Road Networks. *Transportation Research Part C: Emerging Technologies*, Vol. 115, 2020, p. 102618. https://doi.org/10.1016/j.trc.2020.102618.
21. Geroliminis, N., J. Haddad, and M. Ramezani. Optimal Perimeter Control for Two Urban Regions with Macroscopic Fundamental Diagrams: A Model Predictive Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 1, 2013, pp. 348–359. https://doi.org/10.1109/TITS.2012.2216877.
22. Haddad, J. Optimal Perimeter Control Synthesis for Two Urban Regions with Aggregate Boundary Queue Dynamics. *Transportation Research Part B: Methodological*, Vol. 96, 2017, pp. 1–25. https://doi.org/10.1016/j.trb.2016.10.016.
23. Ramezani, M., J. Haddad, and N. Geroliminis. Dynamics of Heterogeneity in Urban Networks: Aggregated Traffic Modeling and Hierarchical Control. *Transportation Research Part B: Methodological*, Vol. 74, 2015, pp. 1–19. https://doi.org/10.1016/j.trb.2014.12.010.
24. Nagle, A. S., and V. V. Gayah. Accuracy of Networkwide Traffic States Estimated from Mobile Probe Data. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2421, 2014, pp. 1–11. https://doi.org/10.3141/2421-01.
25. Du, J., H. Rakha, and V. V Gayah. Deriving Macroscopic Fundamental Diagrams from Probe Data: Issues and Proposed Solutions. *Transportation Research Part C: Emerging Technologies*, Vol. 66, 2016, pp. 136–149.
26. Ambühl, L., and M. Menendez. Data Fusion Algorithm for Macroscopic Fundamental Diagram Estimation. *Transportation Research Part C: Emerging Technologies*, Vol. 71, 2016, pp. 184–197. https://doi.org/10.1016/J.TRC.2016.07.013.
27. Courbon, T., and L. Leclercq. Cross-Comparison of Macroscopic Fundamental Diagram Estimation Methods. No. 20, 2011, pp. 417–426.
28. Saberi, M., H. S. Mahmassani, T. Hou, and A. Zockaie. Estimating Network Fundamental Diagram Using Three-Dimensional Vehicle Trajectories. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2422, No. 1, 2014, pp. 12–20. https://doi.org/10.3141/2422-02.
29. Gayah, V. V., and V. V. Dixit. Using Mobile Probe Data and the Macroscopic Fundamental Diagram to Estimate Network Densities. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2390, No. 1, 2013, pp. 76–86. https://doi.org/10.3141/2390-09.
30. Leclercq, L., N. Chiabaut, and B. Trinquier. Macroscopic Fundamental Diagrams: A Cross-Comparison of Estimation Methods. *Transportation Research Part B: Methodological*, Vol. 62, 2014, pp. 1–12. https://doi.org/10.1016/j.trb.2014.01.007.
31. Kouvelas, A., M. Saeedmanesh, and N. Geroliminis. Enhancing Model-Based Feedback Perimeter Control with Data-Driven Online Adaptive Optimization. *Transportation Research Part B: Methodological*, Vol. 96, 2017, pp. 26–45. https://doi.org/10.1016/j.trb.2016.10.011.
32. Haddad, J., and B. Mirkin. Adaptive Perimeter Traffic Control of Urban Road Networks Based on

MFD Model with Time Delays. *International Journal of Robust and Nonlinear Control*, Vol. 26, No. 6, 2016, pp. 1267–1285. https://doi.org/10.1002/rnc.3502.
33. Haddad, J., and B. Mirkin. Coordinated Distributed Adaptive Perimeter Control for Large-Scale Urban Road Networks. *Transportation Research Part C: Emerging Technologies*, Vol. 77, 2017, pp. 495–515. https://doi.org/10.1016/j.trc.2016.12.002.
34. Mahmassani, H. S., M. Saberi, and A. Zockaie. Urban Network Gridlock: Theory, Characteristics, and Dynamics. *Transportation Research Part C: Emerging Technologies*, Vol. 36, 2013, pp. 480–497. https://doi.org/10.1016/j.trc.2013.07.002.
35. Gayah, V. V., and C. F. Daganzo. Clockwise Hysteresis Loops in the Macroscopic Fundamental Diagram: An Effect of Network Instability. *Transportation Research Part B: Methodological*, Vol. 45, No. 4, 2011, pp. 643–655. https://doi.org/10.1016/j.trb.2010.11.006.
36. Genders, W., and S. Razavi. Using a Deep Reinforcement Learning Agent for Traffic Signal Control. 2016.
37. Li, L., Y. Lv, and F. Y. Wang. Traffic Signal Timing via Deep Reinforcement Learning. *IEEE/CAA Journal of Automatica Sinica*, Vol. 3, No. 3, 2016, pp. 247–254. https://doi.org/10.1109/JAS.2016.7508798.
38. Wei, H., N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li. CoLight: Learning Network-Level Cooperation for Traffic Signal Control. 2019, pp. 1913–1922. https://doi.org/10.1145/3357384.3357902.
39. Liang, X., X. Du, G. Wang, and Z. H. Fellow. *Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks*. 2018.
40. Ni, W., and M. Cassidy. City-Wide Traffic Control: Modeling Impacts of Cordon Queues. *Transportation Research Part C: Emerging Technologies*, Vol. 113, 2020, pp. 164–175. https://doi.org/10.1016/j.trc.2019.04.024.
41. Ni, W., and M. J. Cassidy. Cordon Control with Spatially-Varying Metering Rates: A Reinforcement Learning Approach. *Transportation Research Part C: Emerging Technologies*, Vol. 98, 2019, pp. 358–369. https://doi.org/10.1016/j.trc.2018.12.007.
42. Haddad, J., M. Ramezani, and N. Geroliminis. Model Predictive Perimeter Control for Urban Areas with Macroscopic Fundamental Diagrams. 2012.
43. Henderson, P., R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep Reinforcement Learning That Matters. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2017, pp. 3207–3214.
44. Watkins, C. J. C. H., and P. Dayan. Q-Learning. *Machine Learning*, Vol. 8, No. 3–4, 1992, pp. 279–292. https://doi.org/10.1007/bf00992698.
45. Rummery, G. A., and M. Niranjan. *ON-LINE Q-LEARNING USING CONNECTIONIST SYSTEMS*. 1994.
46. Sutton, R., and A. Barto. Reinforcement Learning: An Introduction. 2018.
47. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-Level Control through Deep Reinforcement Learning. *Nature*, Vol. 518, No. 7540, 2015, pp. 529–533. https://doi.org/10.1038/nature14236.
48. Thrun, S., and A. Schwartz. *Issues in Using Function Approximation for Reinforcement Learning*. 1993.
49. van Hasselt, H., A. Guez, and D. Silver. Deep Reinforcement Learning with Double Q-Learning. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2015, pp. 2094–2100.
50. Van Hasselt, H. *Double Q-Learning*. 2010.
51. Horgan, D., J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver. Distributed Prioritized Experience Replay. 2018.
52. Schaul, T., J. Quan, I. Antonoglou, and D. Silver. Prioritized Experience Replay. 2016.
53. Alsalhi, R., V. V Dixit, and V. V Gayah. On the Existence of Network Macroscopic Safety Diagrams:

Theory, Simulation and Empirical Evidence. *PloS one*, Vol. 13, No. 8, 2018, p. e0200541.