Improving Deep Reinforcement Learning-Based Perimeter Metering Control Methods with Domain Control Knowledge

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.

Keywords: Macroscopic fundamental diagram (MFD); Perimeter control; Deep reinforcement learning (Deep-RL); Domain control knowledge (DCK)

ABSTRACT

Perimeter metering control has long been an active research topic since well-defined relationships between network productivity and usage, i.e., network macroscopic fundamental diagrams (MFD), were shown capable of describing regional traffic dynamics. Numerous methods have been proposed to solve perimeter metering control problems, but these generally require knowledge of the MFDs or detailed equations that govern traffic dynamics. Recently, a study applied model-free deep reinforcement learning methods to two-region perimeter control and found comparable performances to the model predictive control (MPC) scheme, particularly when uncertainty exists. However, the proposed methods therein provide very low initial performances during the learning process, which limits its applicability to real life scenarios. Furthermore, the methods may not be scalable to more complicated networks with larger state and action spaces. To combat these issues, this paper proposes to integrate domain control knowledge (DCK) of congestion dynamics into the agent designs for improved learning and control performances. A novel agent is also developed that builds on the Bang-Bang control policy. Two types of DCK are then presented to provide knowledge-guided exploration strategies for the agents such that they can explore around the most rewarding part of the action spaces. The results from extensive numerical experiments on two- and three-region urban networks show that integrating DCK can: (a) effectively improve learning and control performances for the Deep-RL agents; (b) enhance the agents' resilience against various types of environment uncertainties; and (c) mitigate the scalability issue for the agents.

INTRODUCTION

Transportation researchers and practitioners often use different modeling paradigms to develop, test, and refine traffic control strategies. Microscopic modeling approaches can best represent the reality, but they are not well-suited to urban traffic control due to the complexity of these systems. The network Macroscopic Fundamental Diagram (MFD) has recently emerged as another tool to model urban transportation systems from a regional perspective. Specifically, the MFD leverages the existence of well-defined and unimodal relationships between the average network productivity (e.g., trip completion rate) and average network usage (e.g., accumulation) on homogeneous networks to describe aggregate traffic dynamics. The presence of such relationships has been studied for a long time (*1–4*), but integrating the MFD into a framework that enables aggregate traffic dynamics modeling is a relatively recent achievement (*5*). Since then, extensive MFD related research efforts have been performed, such as MFD estimation (*6–9*), the existence conditions for low-scatter MFDs (*10–13*), network instability and hysteresis phenomena (*14–18*), and others.

MFD-based modeling paradigms have facilitated the development of perimeter metering control (PMC) schemes, i.e., regulating transfer flows to improve the overall network throughput. For single-region networks, the PMC problem was first studied in (*5*) and further investigated in (*19–22*). Numerous research works have also examined PMC for two-region (*23–28*) and multi-region networks (*29–36*). A wide variety of methods have been proposed to solve the PMC problem, and these can be loosely categorized into model-based and data-driven approaches. Model-based methods include proportional-integral based control (*19*, *20*, *32*), adaptive control (*28*, *37*), model predictive control (MPC) (*24*, *25*, *30*, *31*, *38*), and others. In particular, the MPC is an advanced close-loop control scheme that considers the possible discrepancy between the MFDs prediction model and plant (reality). It has been applied extensively in prior works and has realized state-of-the-art control performances. However, by nature of the rolling horizon design, the MPC suffers from low generalizability to new plants because of its sensitivity to horizon parameters (*39*, *40*). More importantly, the successful application of model-based methods is contingent upon relatively accurate modeling of the regional environment dynamics, a problem that is also challenging.

For these limitations, data-driven approaches have received increasing research interests recently. Examples include model free adaptive control (MFAC) (*33*, *34*) and reinforcement learning (RL) methods (*26*, *41–43*). Notably, RL methods could internalize the traffic dynamics and produce control strategies from interactions with the environment, and they have been shown comparable to the MPC (*43*). While remarkable, in the initial period of learning, the RL agents consistently perform worse than when no control is applied. This initial underperformance results from the agents' completely random exploration of the entire action space, which is contrary to how someone with knowledge of the scenario (e.g., domain experts) would explore to intelligently learn about the environment. Hence, the present paper examines how external knowledge can be integrated into the RL agents to improve their learning and control performances. In particular, this paper focuses on the application of the recently-developed C-RL agent in (*43*) and also proposes a novel agent that builds on the Bang-Bang control policy (*5*, *44*) for two- and three-region PMC. Domain control knowledge (DCK) is then presented and integrated within these agents to obtained much improved performances. The DCK initially provides a "warm-start" to the learning processes by defining a set of default actions for the agents that are conditioned on the network congestion level. During the remainder of the training process, it continues to provide the default actions for the agents to explore around at each step to determine their overall control policy. By providing such information, the DCK specifies the most fruitful part of the action space for the agents to enable efficient exploration. A series of explorative experiments are conducted to determine suitable representations for the DCK. The effectiveness of the DCK is demonstrated via extensive numerical experiments on two- and three-region perimeter control problems, where the control outcomes, resilience to environment uncertainties, and scalability to larger problems are comprehensively examined.

The remainder of the present paper is structured as follows. The next section introduces the general formulation of perimeter metering control problems with MFDs. Subsequently, an overview is provided on the application of the C-RL agent to perimeter control, which is followed by the novel Deep-RL agent

proposed herein and the two types of DCK. The following section presents the simulation results, and the final section summarizes the concluding remarks.

PROBLEM FORMULATION

This paper considers a general PMC problem for an urban network composed of a set of N homogenous regions, $R_i, i = 1,2,\cdots,N$. When the regions are not homogenous, network partitioning algorithms could be applied to maintain homogeneity (*45–47*). As such, each region could be modeled with a low-scatter MFD, $f_i(n_i(t))$, which provides the trip completion rate at any regional accumulation $n_i(t)$ observed at time step t. The aggregate modeling of traffic dynamics can then be expressed as follows (*34–36*, *48*):

$$\dot{n}_{ii}(t) = q_{ii}(t) - M_{ii}(t) + \sum_{h\in N_i} u_{hi}(t)M_{hii}(t) \tag{1}$$

$$\dot{n}_{ij}(t) = q_{ij}(t) - \sum_{h\in N_i} u_{ih}(t)M_{ihj}(t) + \sum_{h\in N_i;h\neq j} u_{hi}(t)M_{hij}(t) \tag{2}$$

where n_{ij} and q_{ij} respectively represent the accumulations and traffic demands within R_i destined for R_j, n_{ii} and q_{ii} are defined similarly (with $n_i = \sum_j n_{ij}$), $u_{ih}(t)$ denotes the perimeter controller that specifies the ratio of vehicles allowed to transfer from R_i to R_h where $h \in N_i$ and N_i is the set of neighboring regions to i. The controller values u_{ih} are bounded by $[u_{min}, u_{max}]$ with $0 \le u_{min} < u_{max} \le 1$. $M_{ihj}(t)$ stands for the transfer flow from R_i to R_j through the next region R_h and is calculated by:

$$M_{ihj}(t) = \theta_{ihj}(t)\frac{n_{ij}(t)}{n_i(t)}f_i\big(n_i(t)\big) \tag{3}$$

where $\theta_{ihj}(t) \in [0,1]$ is the route choice term that represents the percentage of transfer flow from R_i to R_j that utilizes R_h (hence $\sum_{h\in N_i}\theta_{ihj}(t) = 1$). Likewise, the internal trip completion flow is given by:

$$M_{ii}(t) = \frac{n_{ii}(t)}{n_i(t)}f_i\big(n_i(t)\big) \tag{4}$$

Similar to (*49*), the networks considered in this work feature an obvious route for each origin-destination pair, in which case the route choice term can be omitted (i.e., $\theta_{ihj} = 1$ for a single region h). The boundary capacity constraints can be omitted as well since they have been shown inconsequential (*30*, *35*, *49*, *50*) and such omission leads to significant computational advantage. Moreover, the effects of perimeter control on boundary queue dynamics are assumed to be marginal in this paper; see (*25*, *42*, *51*, *52*) for more details on the relaxation of this assumption.

The traffic dynamics presented above can be embedded into the controller designs of model-based approaches such as the model predictive control (*24*, *35*). In reality, however, these dynamics are often blended with environment uncertainty that might arise in the MFDs and/or traffic demands. Concretely, the uncertainty in the MFDs and traffic demand are defined as follows (similar to (*43*)):

$$\tilde{f}_i\big(n_i(t)\big) = f_i\big(n_i(t)\big) + \varsigma(t)\cdot n_i(t), \qquad i = 1,\cdots,N \tag{5}$$

$$\tilde{q}_{ij}(t) = \max(q_{ij}(t)\cdot(1+\varepsilon(t),0), \qquad i,j = 1,\cdots,N \tag{6}$$

where $\varsigma(t)$ follows a mean-zero uniform distribution with parameter α and $\varepsilon(t)$ a mean-zero normal distribution with parameter σ. Substituting the corresponding terms in Eqs. (1)-(4) with (5)-(6), one obtains the MFDs plant for model-based approaches or the simulation environment for data-driven methods. In the present work, numerical simulation is conducted for the experiments with the environment built with the traffic dynamics equations in discrete time and with inherent uncertainties in the MFDs and traffic demands, as similar to (*24*, *34*, *35*).

The objective of PMC is to maximize the network throughput, i.e., the cumulative trip completion (CTC). Solving the perimeter control problem then amounts to selecting actions u_{ih} that maximize the CTC while satisfying the traffic dynamics and other constraints (e.g., boundary constraints for the accumulations). Note that, the problem formulation presented above is applicable to all general perimeter control problems. In subsequent sections, two specific instantiations will be studied, i.e., two- and three-region PMC.

METHODOLOGY

This section details the methodology adopted in this paper. The first subsection provides an overview of the C-RL agent applied to perimeter control. The next subsection proposes a novel Deep-RL agent building upon the Bang-Bang control policy. The two types of domain control knowledge (DCK) are then described in the last subsection.

C-RL for Perimeter Metering Control

Reinforcement learning (RL) has long been applied for traffic signal control by the transportation community (*53–56*). However, its application to perimeter metering control is fairly limited, with a few initial attempts in (*26*, *41*, *42*). Nevertheless, the solution methods in these works are still heavily reliant on the full knowledge of system dynamics. (*43*) provides the first examination of completely model-free Deep-RL techniques on two-region perimeter metering control problems, where the continuous agent C-RL has exhibited comparable control performances to the MPC.

The C-RL agent is built upon the model-free off-policy actor-critic learning algorithm Deep Deterministic Policy Gradient (*57*). The agent has an actor that selects continuous real values for the perimeter controllers and a critic that evaluates the selected actions. For the enhancement of scalability to problems with large state and/or action spaces, both the actor and critic are constructed using neural networks. The actor parameters are updated by gradient ascent with the deterministic policy gradient (*58*) while the critic parameters are updated in a similar fashion to Q-learning (*59*). Recent advances that improve learning stability such as experience replay (*60*) and target network are also incorporated to the C-RL agent. In addition, the agent was strengthened with the distributed learning architecture Ape-X (*61*), which helps collect large quantities of experiences for the agent to learn more effectively. In this work, the C-RL agent is adapted for two- and three-region control. Specifically, the tanh activation layer of the C-RL agent is replaced by a linear layer with truncated outputs such that the RL outputs still lie within [-1,1]. This is helpful since larger variations of the actions across consecutive time steps can be achieved whereas the tanh activation restricts such variations. In addition, when the C-RL agent is applied for three-region perimeter control, its actor network is expanded into three dense ReLU layers with 64, 64, and 16 units.

Before applying the C-RL agent, the perimeter control problem is first formulated as a Markov decision process whose major components are state, action, and reward. The state is defined as a collection of accumulations and estimated traffic demands, the latter of which are readily available from historical observations and do not need to be accurate; see (*43*) for the examination of such inaccuracy. The action contains a real value bounded by $[u_{min}, u_{max}]$ for each perimeter controller. The reward guides the agent to achieve the control objective and is hence given by the normalized trip completion in a time step.

Bang-Bang Type Deep-RL Controller (B-RL)

Following the success of Deep-RL on two-region perimeter control problems, this paper presents a novel agent building upon the Bang-Bang control policy (henceforth denoted as B-RL). The Bang-Bang policy was initially proposed in (*5*) and later corroborated in (*42*, *44*) as the optimal form of actions for perimeter control problems. This policy only allows the perimeter controllers to alternate between the minimum and maximum values possible. In general, when the region is uncongested, the maximum controller value is

selected such that the accumulation could approach the critical level to realize higher efficiency. The minimum value is chosen otherwise to prevent the region from exacerbated congestion.

The Bang-Bang policy provides an intuitive yet effective way to manage urban traffic flows at an aggregate level. Building upon this policy and realizing Bang-Bang type control actions, the resulting B-RL agent can achieve promising control performances. As consistent with the Bang-Bang policy, the B-RL agent can only adopt either u_{min} or u_{max} for each perimeter controller u_{ih}. As such, it has an improved level of scalability to larger perimeter control problems over the other discrete control agent previously proposed in (*43*). Other than the action space design, the B-RL agent assumes the same state information as the C-RL agent, i.e., a list of accumulations and estimated traffic demands. Note that, only regional accumulations $n_i(t)$ are included in the state information since they are readily available in an instrumented network, e.g., from loop detectors. In this work, the B-RL is built with the Double DQN learning algorithm and Ape-X distributed architecture (see (*43*) for more detailed description of the algorithmic designs). Additionally, the Q-network of the B-RL agent has three dense ReLU layers of 64, 32, and 16 units.

Domain Control Knowledge (DCK)

Integration of DCK with the C-RL Agent

The C-RL underperforms the no control strategy initially due to its random exploration of the entire action space, which also slows down the learning process. To combat these, this paper proposes knowledge-guided exploration strategies for the agent via the integration of DCK. Specifically, a set of default actions is provided to the agent by the DCK at each step that suggest where to explore around. These default actions are "best-guess" solutions based on general knowledge of regional traffic flow dynamics; thus, they are informed by the physical behavior of the network and how the best control policy might take shape. In this way, the agent performs its random exploration of the action space in a more guided manner since the DCK can specify the most promising exploration direction for the agents. Note that, the "best-guess" default actions not dependent on detailed information about the MFDs or origin-destination patterns.

With these in mind, the intuitions behind the DCK are explained. First, metering should not be imposed for vehicle moving into regions that are very uncongested, i.e., the inbound perimeter controllers should be directly set to the maximum value u_{max} and further exploration is not needed. Second, when a region is very congested, the inbound perimeter controller should be a value close to u_{min} to prevent this region from becoming more congested. Setting the control value to u_{min} in this case will likely cause severe congestion in other regions; thus, the agent should be instructed to explore around actions that are close to (but not necessarily equal to) u_{min}. Finally, in a scenario where the region is nearing congestion (i.e., with accumulation close to the critical value), obtaining a sensible action without relying on advanced control techniques is nontrivial even for domain experts. Hence, in this case the agent needs to perform truly random exploration.

With the above intuitions, the "best-guess" default action values between any pair of neighboring regions i and j are summarized in Table 1. Note that the congestion level of a region i is classified into 3 categories using the critical accumulation n_{ic} and a user-defined parameter $\xi \in [0,1)$:

- uncongested, as represented by $n_i \ll n_{ic}$ and characterized by $n_i < (1-\xi)n_{ic}$;
- near congestion, as indicated by $n_i \approx n_{ic}$ and defined by $(1-\xi)n_{ic} \leq n_i \leq (1+\xi)n_{ic}$; and,
- congested, as implied by $n_i \gg n_{ic}$ and described by $n_i > (1+\xi)n_{ic}$.

The values u_{low} and u_{high} are fixed quantities for the default actions with $0 \leq u_{min} < u_{low} < u_{high} < u_{max} \leq 1$. Moreover, in scenarios absent of DCK instructions (e.g., when the regional accumulations are around the critical values), the default action is set to a random value between u_{min} and u_{max} (e.g., $u_{mid} = (u_{min} + u_{max})/2$ for simplicity). For example, take the congestion situation defined by $n_i \ll n_{ic}, n_j \gg$

n_{jc}. Region i is uncongested, so the inbound perimeter controller u_{ji} is fixed at u_{max}. However, region j is congested, thus its inbound transfer flows are metered by setting u_{ij} to the default value u_{low}.

Table 1. Default actions for the C-RL agent.

	$n_i \ll n_{ic}$	$n_i \approx n_{ic}$	$n_i \gg n_{ic}$
$n_j \ll n_{jc}$	$u_{ij}, u_{ji} \rightarrow u_{max}$	$u_{ij} \rightarrow u_{max}, u_{ji} \sim u_{mid}$	$u_{ij} \rightarrow u_{max}, u_{ji} \sim u_{low}$
$n_j \approx n_{jc}$	$u_{ij} \sim u_{mid}, u_{ji} \rightarrow u_{max}$	$u_{ij}, u_{ji} \sim u_{mid}$	$u_{ij} \sim u_{high}, u_{ji} \sim u_{low}$
$n_j \gg n_{jc}$	$u_{ij} \sim u_{low}, u_{ji} \rightarrow u_{max}$	$u_{ij} \sim u_{low}; u_{ji} \sim u_{high}$	$u_{ij}, u_{ji} \sim u_{mid}$

$\rightarrow$ represents actions are set to this value and further exploration is not needed
$\sim$ represents only default actions are set to this value that the agent can explore around

The original outputs of C-RL lie within $[-1, 1]$ with the default values being approximately zero. However, the perimeter controller values are bounded by u_{min} and u_{max}. To maintain feasibility for the actions, a functional mapping is required to project the original outputs onto $[u_{min}, u_{max}]$. In addition, this mapping needs to project the default output (i.e., zero) into the default action for the agent to utilize the DCK instructions. Consider again the congestion situation defined by $n_i \ll n_{ic}, n_j \gg n_{jc}$, where the default action for u_{ij} is u_{low}. To utilize the DCK, the functional mapping is expected to project -1 to u_{min}, 0 to u_{low}, and 1 to u_{max}. Note that, numerous functional forms can realize the required mapping, and explorative experiments will be conducted to determine a suitable option. Moreover, the explorative experiments will also help specify suitable values for the parameters u_{low}, u_{high}, ξ.

A few clarifications are provided here for the proposed DCK. First, the DCK only specifies the default actions for the agent to explore around but does not limit the range of actions the agent can take. As such, the resulting agent (denoted by *C-RL+DCK*) could still select all possible controller values between u_{min} and u_{max}. To put it another way, the *C-RL+DCK* agent maintains random exploration of the entire action space but searches mainly around the most fruitful section (specified by the DCK) to realize superior learning efficiency. Second, while the default actions are proposed only for a pair of neighboring regions, they can be applied in scenarios with multiple pairs of regions, i.e., multi-region perimeter metering control, as will be shown in due course. Third, the regional congestion levels are derived from comparisons with the critical accumulations; thus, the resulting agent is not strictly model-free. Fortunately, estimates of the critical accumulations can be conveniently obtained from historical traffic data (*9*). While such estimation might be prone to errors due to multivaluedness, instability, and hysteresis phenomena (*14–16*), this is not a fundamental issue since the estimations are only utilized to provide a warm-start for the agent which will conduct its own learning later on. Moreover, robustness tests will be conducted in this work to demonstrate the resilience of the *C-RL+DCK* agent against such estimation errors. Finally, note that the DCK guidance is embedded in the whole learning process of the agent by providing default actions at each action-taking step. The default actions specified by the DCK not only provide a superior starting point for the agent to initiate its learning course but also help elevate its exploration efficacy during its whole course of learning, as will be demonstrated in the results.

Integration of DCK with the B-RL Agent

Similar as above, the DCK proposed here provides the B-RL agent with a series of default actions that can lead to the most efficient exploration of the action space. However, since the B-RL agent assumes a Bang-Bang type action space, the default actions differ from those for the C-RL agent; see Table 2.

Table 2. Initial actions for the B-RL agent

	$n_i \ll n_{ic}$	$n_i \approx n_{ic}$	$n_i \gg n_{ic}$
$n_j \ll n_{jc}$	$u_{ij}, u_{ji} \rightarrow u_{max}$	$u_{ij} \rightarrow u_{max}, u_{ji} \in \{u_{min}, u_{max}\}$	$u_{ij} \rightarrow u_{max}, u_{ji} \sim u_{min}$
$n_j \approx n_{jc}$	$u_{ij} \in \{u_{min}, u_{max}\}, u_{ji} \rightarrow u_{max}$	$u_{ij}, u_{ji} \in \{u_{min}, u_{max}\}$	$u_{ij} \sim u_{max}, u_{ji} \sim u_{min}$
$n_j \gg n_{jc}$	$u_{ij} \sim u_{min}, u_{ji} \rightarrow u_{max}$	$u_{ij} \sim u_{min}; u_{ji} \sim u_{max}$	$u_{ij}, u_{ji} \in \{u_{min}, u_{max}\}$

$\rightarrow$ represents actions are set to this value and further exploration is not needed
$\sim$ represents initial actions are set to this value with a high probability κ
$\in$ represents actions are chosen by the agent via truly random exploration

As Table 2 shows, the regional congestion levels are defined in the same way as for C-RL with the parameter ξ. The intuitions behind the default actions also resemble those for the C-RL agent. Specifically, vehicles should be allowed entries to uncongested regions, so the inbound controller should be the maximum value u_{max} (denoted by "$\rightarrow$"). For rather congested regions, the inbound controller should be set to the minimum value u_{min} with a high probability, κ (denoted by "$\sim$"). The κ value should not be fixed to 1 as this might cause worsened congestion in other regions. Also, this could lead to insufficient exploration for the agent, which might diminish its learning ability. On the contrary, the probability κ should be at least 0.5 since otherwise the agent is acting against domain expertise. Lastly, when the region is about congested, the agent chooses its action via truly random exploration (denoted by "$\in$"). Such random exploration is based on the $\epsilon-$ greedy strategy, i.e., the action with the maximum Q-value is chosen with probability $1-\epsilon$ and a random action otherwise. Moreover, to ensure effective application of the DCK for the resulting agent (denoted as *B-RL+DCK*), explorative experiments will be performed to decide on the values for κ, ξ.

EXPERIMENTS

In this section, numerical experiments are conducted on two- and three-region perimeter control problems. The Deep-RL agents (i.e., *B-RL*, *C-RL*, and the associated DCK agents) are applied for control to these problems, and the effectiveness of DCK is evaluated in terms of the learning performances and control outcomes. For comparison purposes, two methods that are not learning-based, i.e., model predictive control (MPC) and no control (NC), are also applied for control. The MPC is an advanced control scheme that builds upon relatively accurate modeling of the environment dynamics. Its closed-loop structure renders it applicable to scenarios with discrepancies between the prediction model and plant (reality). Over the past decade, the MPC scheme has been extensively applied to perimeter control problem and has achieved state-of-the-art control performances; see (*25*, *30*, *31*, *35*, *36*). In addition, see (*24*) for computational details and (*43*) for an overview of the MPC method. In the present work, the MPC method is implemented according to (*24*, *35*) with the parameters to be presented shortly. The NC method is included as a baseline since it simulates scenarios where no perimeter control is enforced. This strategy selects u_{max} for all perimeter controllers and generally provides the lower-bound control performances that should not be penetrated.

It should be noted that, while the optimal perimeter control policy has been shown in the form of Bang-Bang (*44*), the Bang-Bang control policy itself is not an effective method and thus not included for comparison herein. The reasons are twofold. For one, under the Bang-Bang policy, most vehicles will be denied entry to a congested region as the perimeter controller will be set at the minimum value. Then, in an urban network comprised of more than one region, transfer flows will be strictly limited when all regions are congested. As a result, vehicles will be held waiting within the origin regions and congestion cannot be well distributed (or dissipate) over the whole network. For another, the effectiveness of the Bang-Bang policy is contingent upon accurate critical accumulation information as it acts upon the congestion status of the region. With even slight underestimation of the regional production, the Bang-Bang policy will impose more-than-necessary restrictions on the transfer flows, which would adversely affect the total trip completion in the network. Therefore, to achieve sufficient perimeter control efficacy, advanced control schemes like the MPC are required. In this work, though, Deep-RL methods are utilized since they do not depend on accurate knowledge of the environment; see (*43*) for more discussions on this.

In all experiments, the boundary values of $u_{min} = 0.1, u_{max} = 0.9$ are adopted to consider the practical implementations of perimeter control. Concretely, a complete prohibition ($u_{min} = 0$) of transfer flows can rarely be enforced in real life, despite its theoretical feasibility. On the other hand, while $u_{max} = 1.0$ can potentially lead to higher total trip completion since more vehicles can cross the regional border at each time step, such accommodations can hardly be achieved in practice due to factors like lost time of the traffic signals. Further, note the selection of these boundary values is pretty common in the literature; for example, see (*24*, *30*, *31*, *35*, *49*) and others. The duration of a time step in the control period is set to $\Delta t = 60s$, which is the assumed cycle length of traffic signals along the cordon boundary that implement perimeter control, as consistent with (*24*, *31*, *34*). Moreover, as described in the problem formulation, the MFDs plant or simulation environment is expected to exhibit uncertainty in the MFDs and/or traffic demands. In the present paper, the uncertainty level of $\sigma = 0.2, \alpha = 0.2$ is considered.

Two-Region Perimeter Control

The experiment setup of two-region perimeter control is first introduced, where the main control scenario is explained on which different methods are compared. Under the main scenario, explorative experiments are then performed to determine suitable representations for the DCK. Subsequently, control results on the main scenario using different methods are described, followed by a series of robustness tests.

Experiment Setup

An urban network comprising two homogenous regions is considered here; see Fig. 1(a). Each region is modeled by a well-defined MFD with R_1 assuming the one observed in Yokohama, Japan (*9*). Note that, the MFDs adopted in this paper assume a piecewise functional form with linear for extreme congestion and third-order polynomial otherwise. This functional form has been shown more accurate and realistic in terms of macroscopic traffic dynamics representations (*43*, *62*). The inner region R_2 simulates a city center and is modeled by a smaller MFD, as in Fig. 1(b). The scatter therein represents the uncertainties in the MFDs (with $\alpha = 0.2$ as in Eq. (5)), which indicate the mismatch between the perceived MFDs and the true ones in the environment. The critical accumulations are $n_{1c} = 8{,}241$ veh and $n_{2c} = 4{,}120$ veh. The estimated traffic demand profile is presented in Fig. 1(c), where the overall demand to the simulated city center R_2 is much higher than to the periphery R_1, as characteristic of traffic conditions during morning peaks. Similarly, the scatter in Fig. 1(c) implies the differences between the true and estimated traffic demands (with $\sigma = 0.2$ as in Eq. (6)), which simulate the temporal fluctuations of traffic demands across different days. The initial accumulations are assumed as $n_1(0) = 6{,}000$ veh and $n_2(0) = 5{,}000$ veh that separately represent an uncongested periphery and a congested city center.

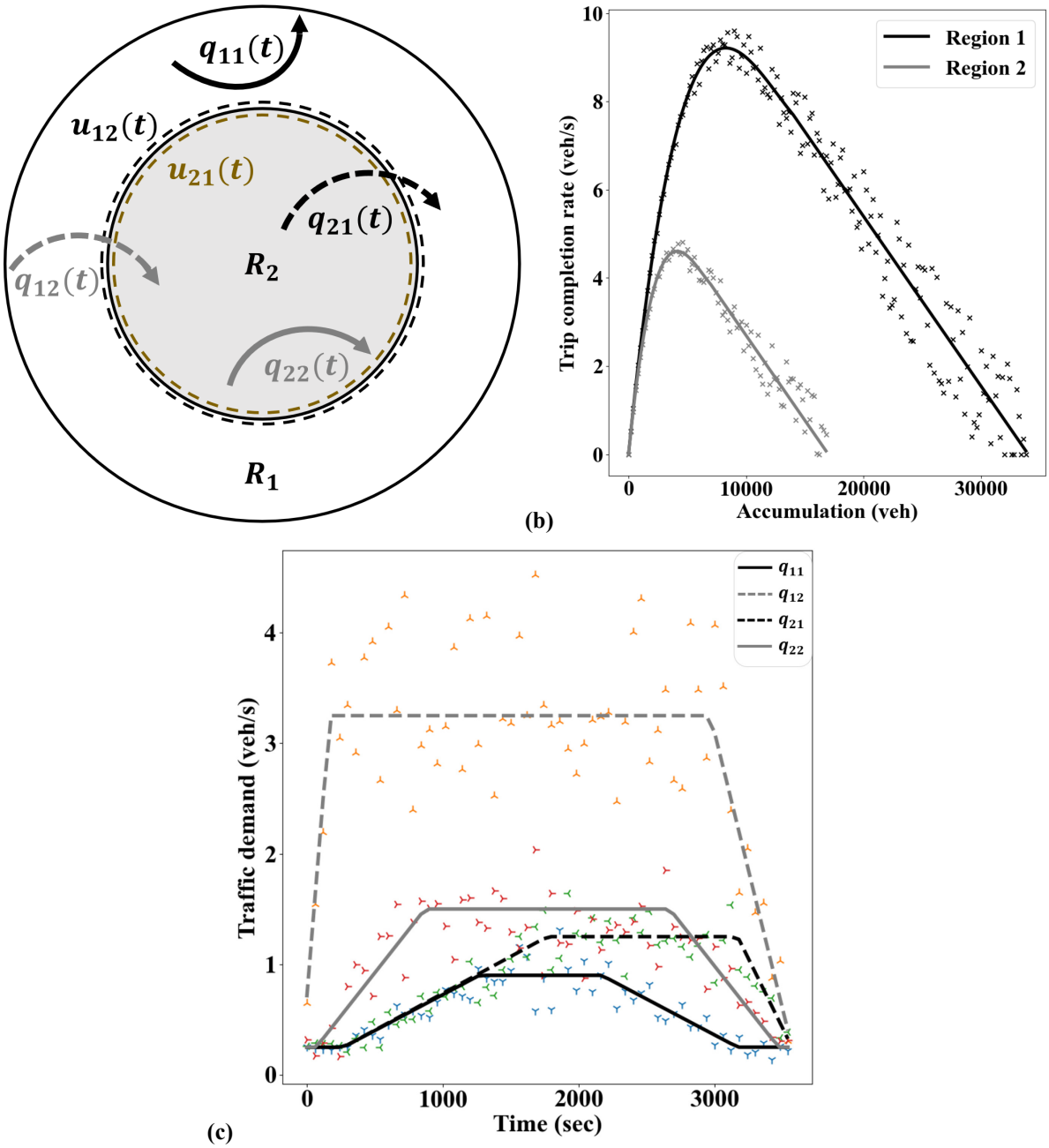

Fig. 1. (a) a two-region urban network; (b) the associated MFDs; (c) estimated traffic demand profile. The scatter in (b) and (c) respectively represents the uncertainties in the MFDs and estimated traffic demands.

For the two-region control problem considered here, the MPC is implemented as per (*24*) with both prediction and control horizons of 20, i.e., the MPC method considers the traffic dynamics for 20 time steps into the future and controls for the 20 steps. The selection of the control horizon aims to provide the best possible MPC control performance. The state for all agents consists of 2 regional accumulations n_i and 4 estimated traffic demands $q_{ij}, i, j = 1,2$. All components of the state are scaled into $[0,1]$ via a division by their respective maximum values. For the DCK agents, an indicator variable denoting the network congestion level is also appended to the state so they can start learning with the default actions specified by the DCK. The outputs of all four Deep-RL agents are two real values with one for each of the two perimeter

controllers. For the *C-RL* and *C-RL+DCK* agents, the outputs could be any value between u_{min} and u_{max}; whereas those of the *B-RL* and *B-RL+DCK* agents are either u_{min} or u_{max}. All four agents receive the normalized trip completion at a time step as the reward, which helps guide them to realize the maximum possible total trip completion. At each training iteration, the demand profile (with uncertainty) in Fig. 1(c) along with the initial accumulation information is fed to the agents step by step for them to take actions. In addition, all neural networks (i.e., the actor and critic networks of *C-RL* and *C-RL+DCK* as well as the Q-network of *B-RL* and *B-RL+DCK*) are constructed as simple multi-layer perceptron, a structure common adopted in the literature (*57*, *61*). Further, the DCK agents adopt the same hyperparameters as the original agents so that a fair comparison can be established. For more details about the training setups and procedure, the reader could refer to Algorithms 1 and 2 in (*43*). In total, six methods are compared for their ability to perform perimeter control: *B-RL, B-RL+DCK, C-RL, C-RL+DCK, MPC, NC*.

Explorative Results

As alluded previously, the effectiveness of the DCK might be affected by their specific representations, i.e., the functional form for the default action mapping and $u_{low}. u_{high}, \xi$ values for *C-RL+DCK*, and the κ, ξ values for *B-RL+DCK*. Hence, this section examines the impacts of these decisions via a set of explorative experiments on the main control scenario to find suitable representations.

DCK representation for the C-RL agent

The functional mapping for the default actions is studied first. To this end, the default action values are preliminarily set to $u_{low} = 0.3, u_{high} = 0.7$ while the ξ value is set to 0.05. It is worth reiterating that the function form is supposed to map the RL outputs (which lie within $[-1,1]$ with the default value of 0) to $[u_{min}, u_{max}]$ with the default value being the default action. Different functional forms considered in this work are presented in Fig. 2(a) for a scenario with the default action u_{low}, in which case -1 is mapped to $u_{min} = 0.1$, 0 to $u_{low} = 0.3$, and 1 to $u_{max} = 0.9$. The DCK assuming each functional form was integrated with the C-RL agent to perform perimeter control on the main scenario, and their performance curves, which represent the cumulative trip completion (CTC) during the peak hour over training iterations of the agent, are presented in Fig. 2(b). The agents were trained for 250 iterations, where each iteration includes a batch of episodes (i.e., a complete sequence of states-action-reward pairs). The two benchmarking methods, MPC and NC, are run 5 times, and the median performances are reported. Note that, the original curves of the agents are rather noisy due to high environment uncertainties in the MFDs and traffic demands, so a moving average of window size 5 is used to smooth out the curves. As shown, the overall learning performances of the *C-RL+DCK* agents are relatively similar across different functional forms, indicating the robustness of the proposed DCK against functional mappings. For computational simplicity of the gradient, which might impact the training processes of the agents, the quadratic functional form is selected henceforth.

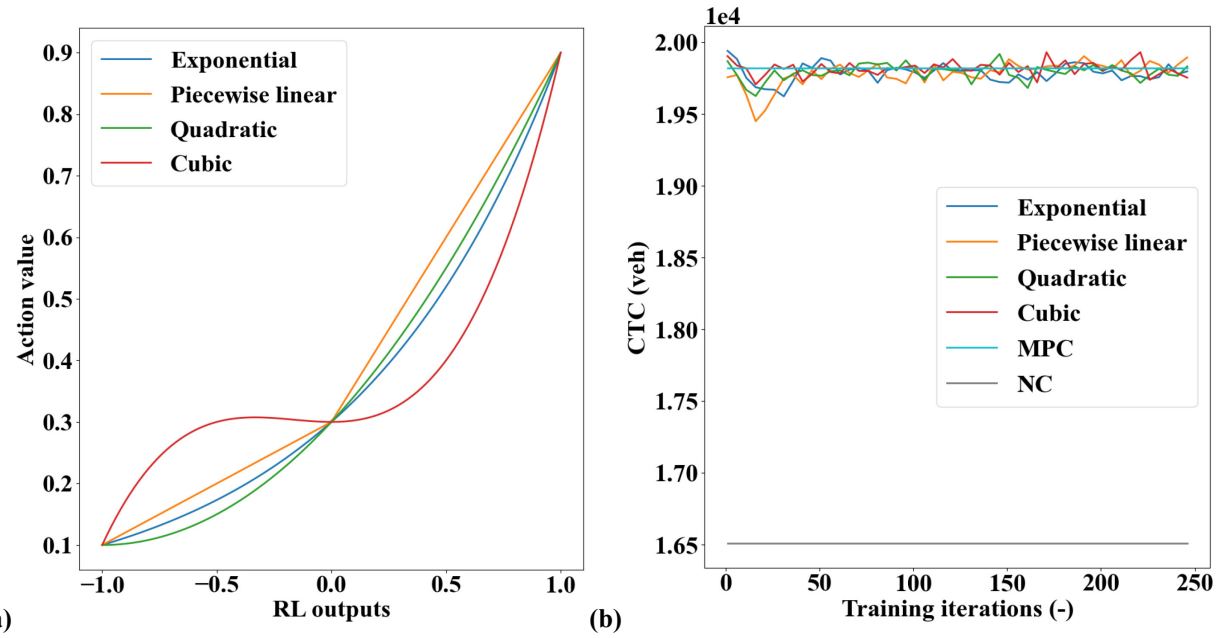

Fig. 2. (a) different functional forms for the DCK; (b) performance curves with moving average.

To evaluate the impacts of the default actions, a range of u_{low} values from 0.20 to 0.40 with 0.05 increments are tested. This range is considered since u_{low} should be closer to u_{min} compared with u_{max}. Also, the constraint $u_{low} + u_{high} = 1.0$ is maintained to reduce the space of possible default action values that need to be tried. In addition, the quadratic functional form is utilized as demonstrated above and ξ is still set at 0.05. The quadratic functions with different u_{low} values are shown in Fig. 3(a). Notice that some quadratic functions exhibit action values lower than u_{min}. A similar situation arises when exploration of the agent leads to infeasible actions (i.e., smaller than u_{min} or larger than u_{max}). To address this, all actions selected by the agent are truncated into $[u_{min}, u_{max}]$ before implementation to maintain feasibility for the perimeter controllers. The DCK with different u_{low}values is then integrated into the C-RL agent to perform perimeter control, and the performance curves are presented in Fig. 3(b). Moreover, the CTC realized by the agent at the first iteration is plotted against the u_{low} values in Fig. 3(c) to demonstrate how well the default actions work. Note that, the CTC values at the first iteration are purely dependent on the DCK since the agent has not been trained and relies on the DCK to take the default actions. The default actions, on the other hand, are determined by the u_{low} values (since $u_{min}, u_{max}, u_{mid}$ are fixed and u_{high} can be decided by u_{low}). Therefore, Fig. 3(c) could be utilized to evaluate the impacts of u_{low} values on the DCK in terms of the starting point it provides to the C-RL agent.

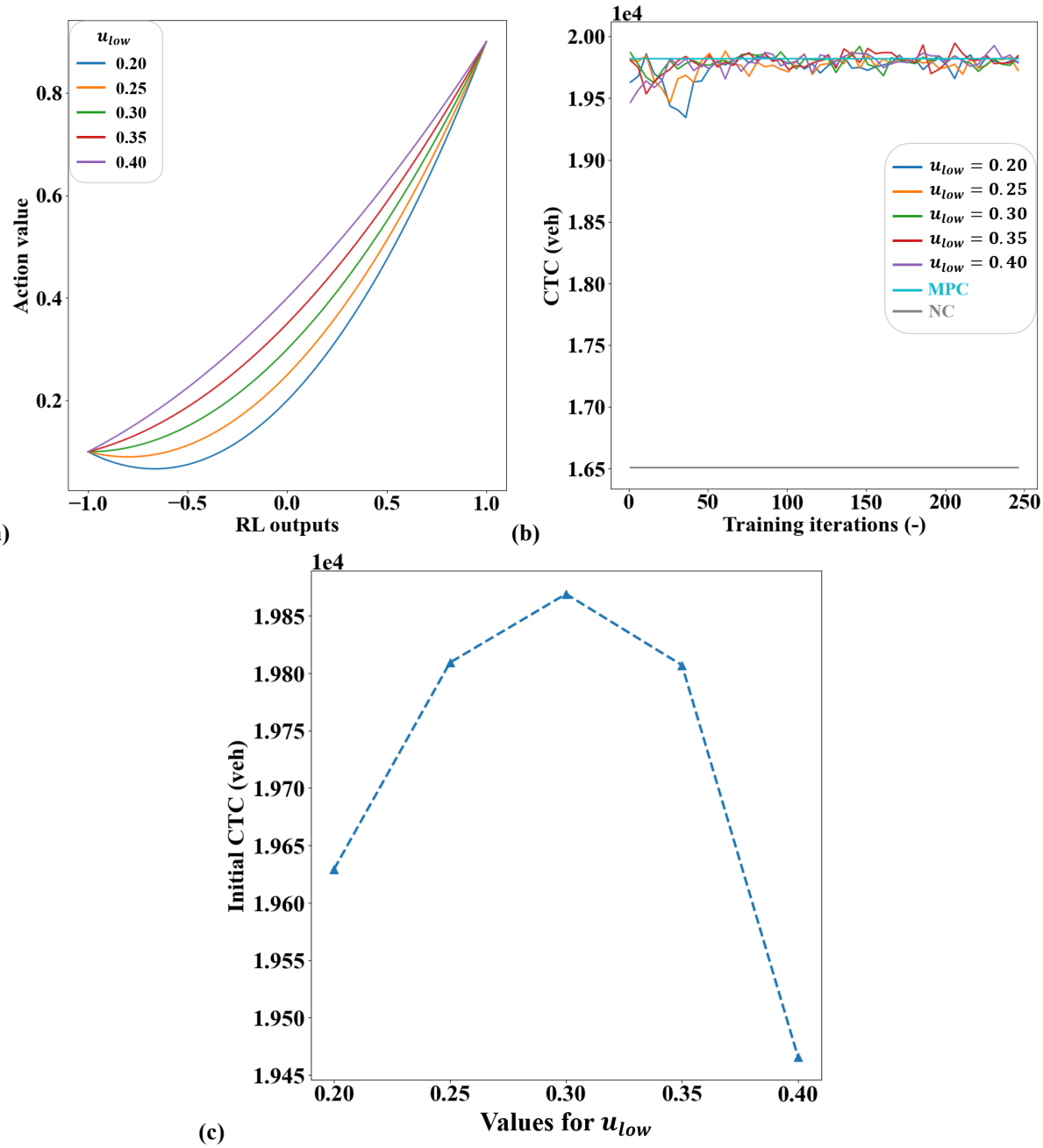

Fig. 3. (a) Quadratic functions with different u_{low} values; (b) performance curves with moving average; (c) CTC values achieved by *C-RL+DCK* agent at the first iteration against the u_{low} values.

As can be observed in Fig. 3(b), the learning curves tend to fluctuate less as the u_{low} value increases. When the u_{low} value becomes greater than 0.25, the amount of fluctuation in the learning curves stays relatively constant. The CTC values achieved by the agent at the first iteration are comparatively low for small (< 0.25) and large (> 0.35) u_{low} values, which are well within expectation. Specifically, DCK with small u_{low} values would likely cause the resulting agent to limit transfer flows to an excessive extent, thus leading to congestion that cannot fully dissipate. Contrarily, DCK with large u_{low} values would provide to the agent default actions that deviate from u_{min}, u_{max} and approach u_{mid}, making the exploration strategy a lot similar to completely random exploration. As a result, the agent ends up exploring much of the action space that is not fruitful enough and fail to perform the desired effective metering. Hence, the CTC values

decrease rapidly as u_{low} exceeds 0.35. From these observations, the default action values are set to $u_{low} = 0.3, u_{high} = 0.7$ for the DCK.

To determine a suitable value for the ξ parameter that defines the classifications of regional congestion levels, a series of experiments have been conducted using the *C-RL+DCK* agent for control while the ξ value changes from 0.05 to 0.25 with 0.05 increments. The quadratic function form is utilized, and the default actions are $u_{low} = 0.3, u_{high} = 0.7$. Intuitively, as ξ decreases, fewer accumulation values can be classified into $n_i \approx n_{ic}$. As a result, the congestion levels are more accurately defined, which renders the DCK more useful. Note that, while the critical accumulations used to derive the congestion levels are assumed to be precise here, the estimation errors will be explicitly examined in subsequent sections. Using different ξ values for the congestion level definition, performance curves of the C-RL+DCK agent are presented in Fig. 4(a). To compare the impacts of different ξ values in a clearer manner, summary statistics of the performance curves are provided in Fig. 4(b), where the primary axis denotes the mean value of CTC while the secondary axis indicates the standard deviation (S.D.). Naturally, higher values on the primary axis suggest higher network throughput and lower values on the secondary axis imply better control stability. Based on these two criteria, the value of 0.05 is selected for the ξ parameter as it exhibits high network throughput and superior control stability.

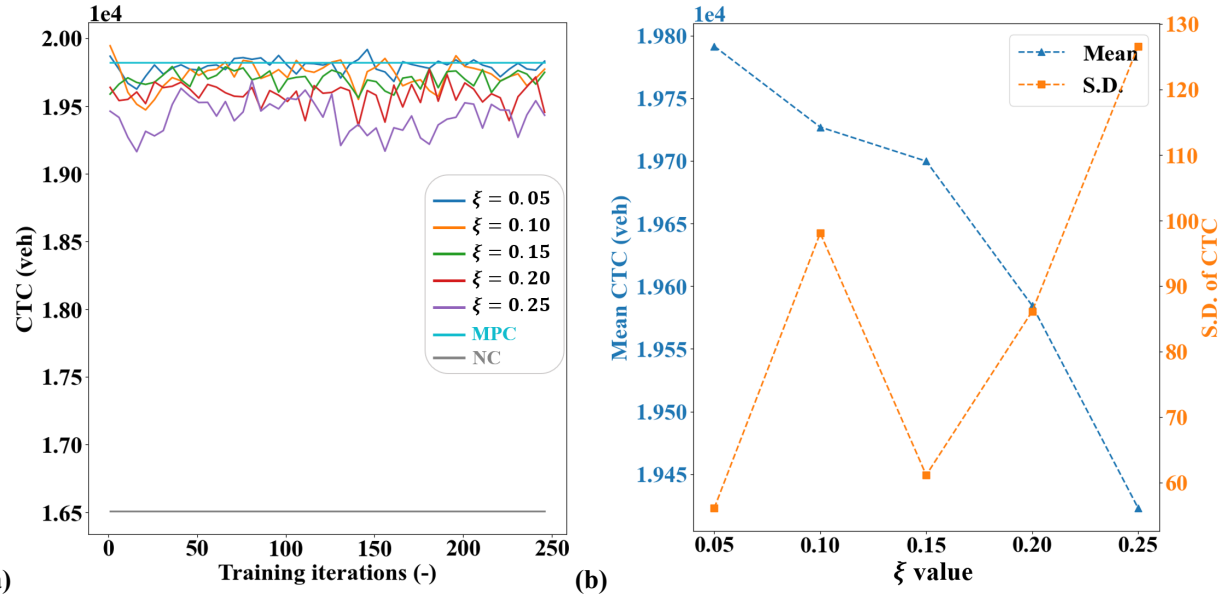

Fig. 4. (a) Performance curves with moving average; (b) summary statistics.

DCK representation for the B-RL agent

The parameter κ could indicate the level of guidance provided to the B-RL agent or the level of confidence domain experts are about the recommended actions. As such, only values that are ≥ 0.5 are considered for κ. In addition, setting $\kappa = 1.0$ would potentially cause severe congestion in other regions or even diminish the agent's ability to explore and learn about the environment. Therefore, values from 0.5 to 0.9 are selected and the DCK with each value is infused into the B-RL agent for perimeter control. The ξ value is set to 0.05. The performance curves as well as the summary statistics are respectively presented in Fig. 5(a) and (b). As can be observed, with the κ value approaching 0.9, both the network throughput and control stability improve. On the contrary, as the κ value decreases towards 0.5, the learning and control performances of the B-RL+DCK agent deteriorate. This is expected since $\kappa = 0.5$ means the agent is alternating between u_{min} and u_{max} in a completely random manner, which then implies insufficient guidance from the DCK. Therefore, the value of 0.9 is chosen for the κ parameter.

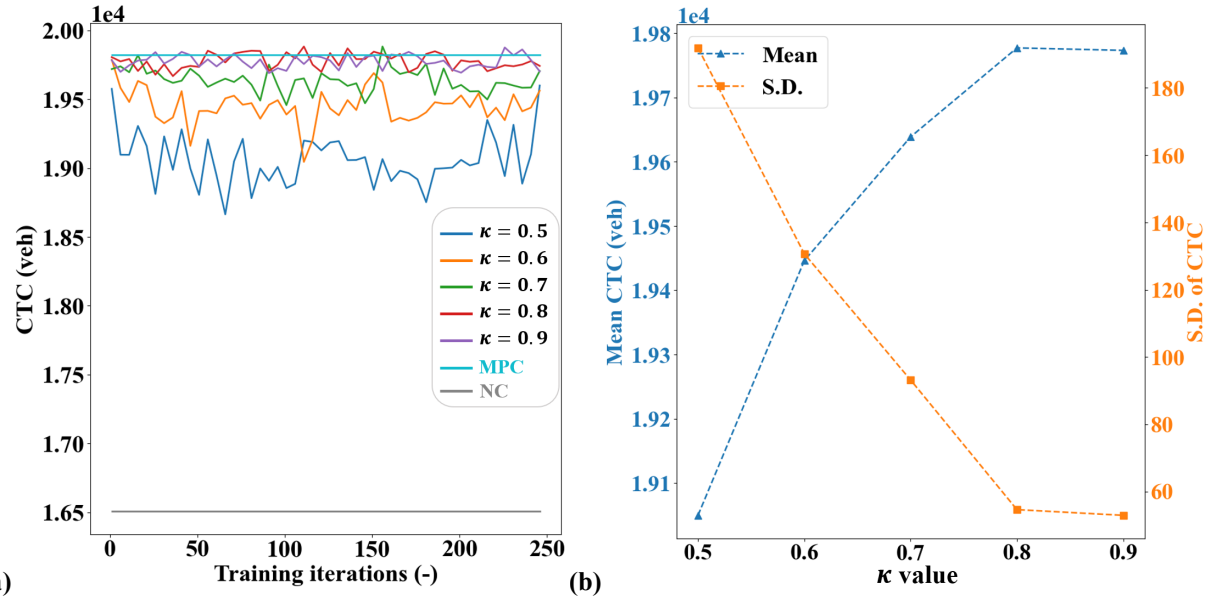

Fig. 5. (a) performance curves with moving average; (b) summary statistics

The ξ parameter for the B-RL agent is examined in this last set of explorative experiments with values from 0.05 to 0.25 considered. The κ value is set to 0.9. The performance curves and the summary statistics are separately presented in Fig. 6(a) and (b). Similar to Fig. 4, the value of 0.05 is associated with high network throughput and great control stability. Therefore, 0.05 is selected to derive the congestion levels for the B-RL agent.

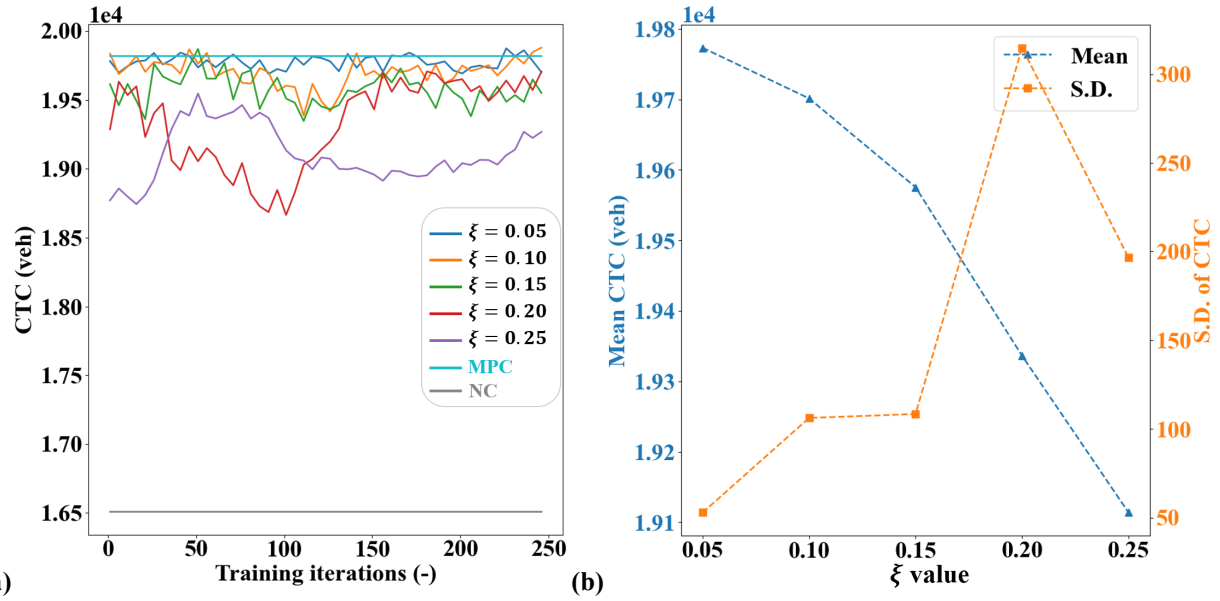

Fig. 6. (a) performance curves with moving average; (b) summary statistics

To summarize, the explorative results have shown that: (a) for the C-RL agent, the DCK should adopt the quadratic functional mapping and the parameters should be $u_{low} = 0.3, u_{high} = 0.7, \xi = 0.05$; (b) for the B-RL agent, the parameters should be $\kappa = 0.9, \xi = 0.05$. These parameterization settings are

embedded into the DCK representations, which are then integrated with corresponding Deep-RL agents for control. To demonstrate the effectiveness of the DCK, the control results will be presented subsequently.

Control Results

Integrating the DCK with above parameterization settings, performance curves of the six control methods are provided in Fig. 7. The individual curves represent the evolution of cumulative trip completion (CTC) with respect to the training iterations using different solution methods. A total of four random seeds were used to train the Deep-RL agents and each seed leads to a slightly different performance curve. The darker lines in Fig. 7 denote the median performances across random seeds whereas the shaded areas represent performance bounds. Similarly, the NC and MPC are run multiple times to obtain their performance bounds; however, their performances are relatively invariant as they are not learning-based methods.

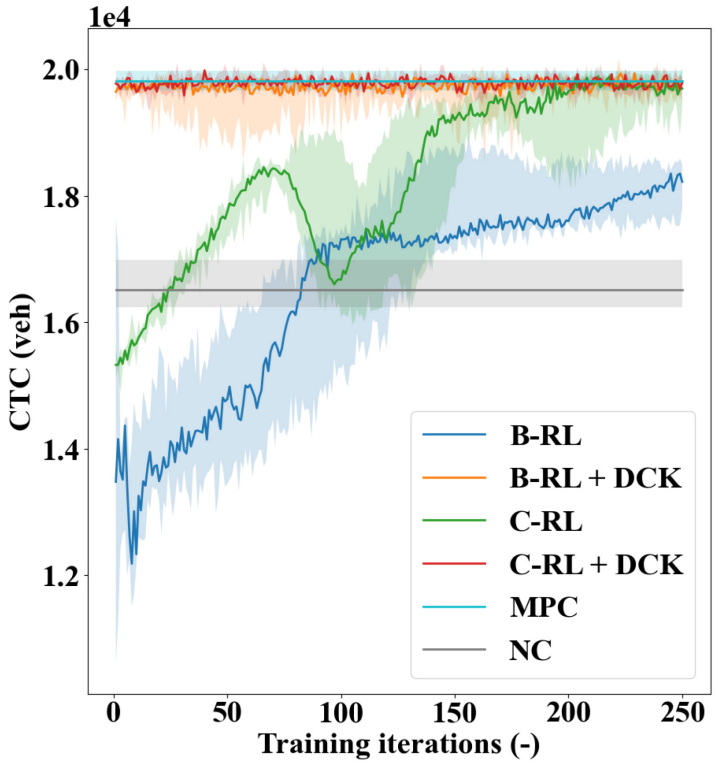

Fig. 7. Performance curves of the six control methods on the main control scenario.

As can be observed in Fig. 7, the NC method provides the lower-bound performance in terms of the final CTC values, which is expected since unrestricted vehicle influx into the already congested city center will only exacerbate the congestion therein. In contrast, the MPC achieves a much higher CTC than the NC, indicating both the necessity and advantage of perimeter control. The C-RL agent can steadily learn and its performance is comparable with the MPC upon convergence. The B-RL agent, however, fails to compete with the MPC although it can conduct learning to some extent. Note that, the CTC values realized by both agents in the early period of learning are worse than the NC, which will likely not be tolerated in real-world implementations. Comparatively, the DCK agents initiate the learning processes from a high-performance point, as attributed to the advantageous default actions specified by the DCK. Importantly, the DCK can help improve the learning processes of the C-RL agent without impacting its final performances. For the B-RL agent, the DCK can elevate both its initial and final control performances. Additionally, the DCK agents stabilize at the best performances much earlier than the original agents. These observations

manifest the effectiveness of the DCK. While promising, these results are not unexpected since the DCK can specify the most fruitful part of the action spaces for the agents to explore. It is worth highlighting that, the DCK is not only providing a better starting point for the agents. More importantly, it is infusing domain expertise-based exploration strategy into the learning procedure of the agents in the form of default actions at each step, which is what truly contributes to the improved performances for the agents.

To examine the effectiveness of the DCK in greater detail, control outcomes of the six methods are visualized in Fig. 8 and Fig. 9, respectively, for the evolutions of accumulations and controller values of u_{12}. Since the traffic demands to R_1 are comparatively small while the region has a larger MFD, transfer flows bounded for R_1 are not limited, i.e., $u_{21} = u_{max}$ for all methods. This is thus not included in the discussions here. As shown in Fig. 8, under the NC strategy, the accumulation in R_1 consistently decreases, which corresponds to the ever-increasing accumulation in R_2 since the transfer flows are not metered. As a result, reduced vehicle presence in R_1 and worsened congestion in R_2 lead to a rather low throughput. On the other extreme, the B-RL agent limits the transfer flows to an excessive extent, as revealed in Fig. 9(a). Hence, region 1 ends up in a congested state while region 2 suffers from low efficiency due to unutilized capacity. In comparison, the other four methods are able to maintain the regional accumulations around the critical values that are associated with the maximum throughput. This observation shows that these methods can indeed perform perimeter control in an optimal manner. Furthermore, Fig. 8 shows a high level of resemblance between the MPC and DCK agents regarding the resulting evolutions of accumulations, which is suggestive of a high level of comparability between these methods. This is particularly demonstrative of the DCK effectiveness since the DCK can not only improve the initial performances for the agents but also elevate their final performances to a level directly comparable with the state-of-the-art MPC method.

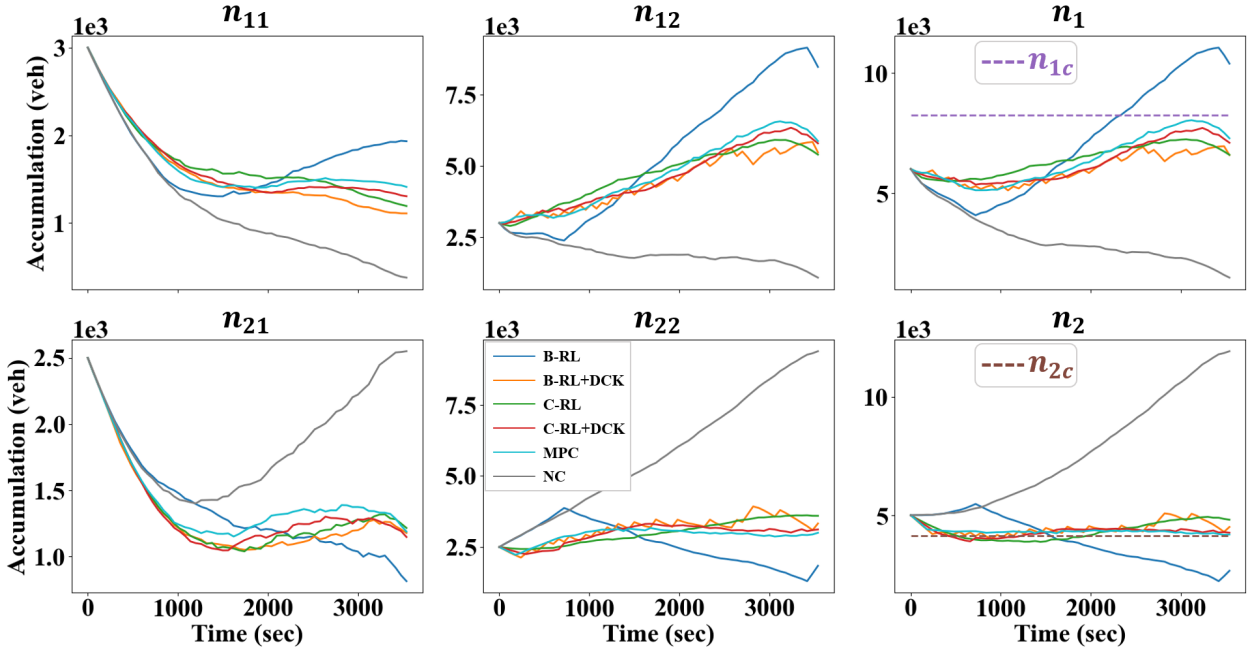

Fig. 8. Evolutions of accumulations for different control methods.

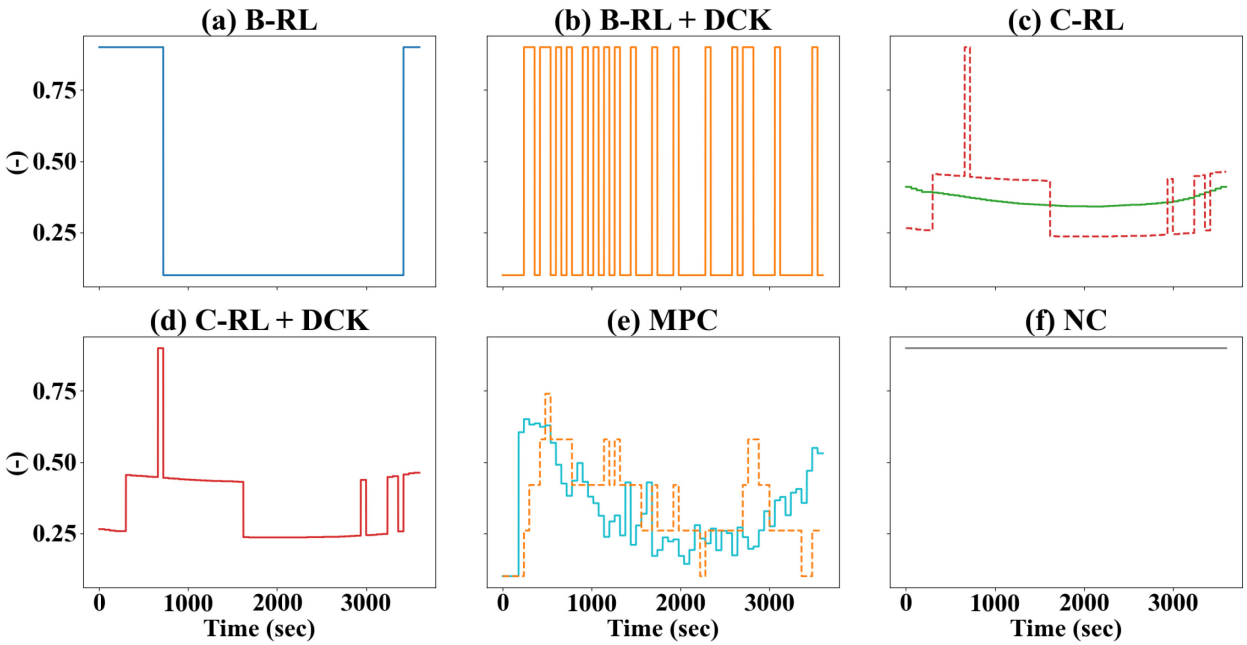

Fig. 9. Controller values of u_{12} over time by each method.
In (c), control action of C-RL is overlaid by that of *C-RL+DCK* in red dash lines.
In (e), control action of MPC is overlaid by that of *B-RL+DCK* with moving average in orange dash lines.

A few additional observations can be made from Fig. 9. First, the DCK can be utilized to facilitate abrupt changes in the policy. To see this, notice how the policies differ between C-RL and *C-RL+DCK* in Fig. 9(c). Concretely, during 300-1600s, R_2 accumulation satisfies $n_2 \approx n_{2c}$, so the *C-RL+DCK* searches around 0.5 (u_{mid}) and ends up with controller values larger than the C-RL. Similarly, during 1600-3200s, the agent generally has smaller controller values than the C-RL as it searches around 0.3 (u_{min}). The C-RL features a smooth control policy, which leads to ease of practical implementations of perimeter control. However, a policy that allows abrupt changes might have more potential to better handle the fluctuations in traffic conditions. Second, Fig. 9(e) reveals the similarity between the policies of MPC and *B-RL+DCK* during implementation time since field implementations of perimeter control often utilize moving average to better execute the desired control (*24*). This similarity then showcases how the DCK could help an agent that fails to learn a good policy originally to perform effective perimeter control.

Robustness Tests

To further demonstrate the DCK effectiveness and the resilience of the DCK agents against environment uncertainties, this section considers two more experiment scenarios with different types of uncertainties.

Measurement noise

Building upon the main scenario with uncertainties in the traffic demands and the MFDs, this test examines the resilience of the DCK agents against environment measurement noise on the accumulations. Concretely, the measurement noise considered here is in the form of $\tilde{n}_{ij}(t) = n_{ij}(t) + \mathcal{N}(0, \delta^2)$ where $\tilde{n}_{ij}(t)$ is the measured accumulation in the environment, $n_{ij}(t)$ is the perceived accumulation by different methods, and $\mathcal{N}(0, \delta^2)$ represents a normal distribution with scale parameter δ. This type of uncertainty simulates the possible malfunction of sensors (e.g., loop detectors) that may lead to inaccurate identification of vehicles. Under different δ values, the DCK agents are utilized for perimeter control, and their control gains (in terms of CTC) are compared with the MPC and NC methods; see Fig. 10. As can be observed, both DCK agents

can consistently outperform the MPC irrespective of the noise level. Also, the performance margin roughly increases with the noise level. This indicates the MPC becomes less effective for perimeter control with increased modeling uncertainty, which is reasonable since the MPC builds upon relatively accurate modeling of system dynamics. With increased discrepancies between the prediction model and plant, the MPC is likely to suffer from deteriorated control performances as it receives incorrect information from the environment. In contrast, the DCK agents can internalize the system dynamics without dependence on the accurate accounts of the environment, which explains the resilience to different levels of measurement noise. Further, the MPC considers only a segment of the future conditions (as bounded by the prediction horizon), whereas the DCK agents can account for the entire simulation period via continuous interactions with the environment. Finally, note that it is not atypical for data-driven methods to exhibit superior performances to the MPC, especially with high levels of environment uncertainty; see (*33*, *34*, *43*) for more discussions and examples on this regard.

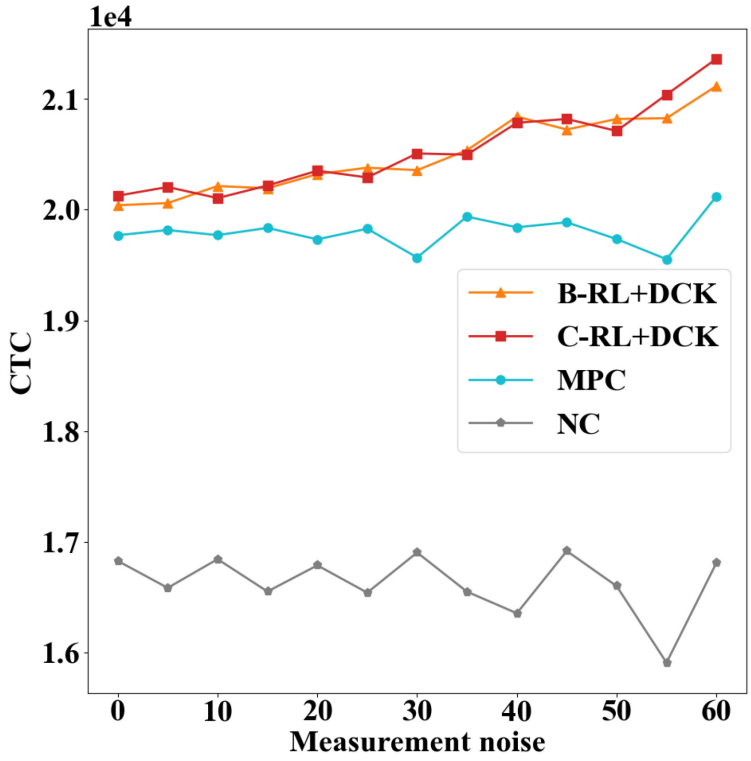

Fig. 10. Control gains of different methods under each level of measurement noise.

Estimation errors of the critical accumulation

As hinted previously, the DCK is designed based on definitions of the regional congestion levels that require the critical accumulation information, whereas estimating the critical accumulations might be prone to errors due to multivaluedness, instability, and hysteresis phenomena (*14–16*). Hence, this test examines whether the DCK can still be effective when the critical accumulations are inaccurately estimated and whether the DCK agents are resilient to such estimation errors. To this end, estimation errors ranging from -20% to 20% are tested. For example, an estimation error of -20% means the DCK classifies accumulation values of $n_i > (1+\xi)\cdot n_{ic}\cdot(1-20\%)$ as $n_i \gg n_{ic}$, i.e., the regional production is severely under-estimated. Contrarily, positive estimation errors suggest that the regional production is over-estimated. Note that, the estimation errors apply to all regions consistently, i.e., the regional productions are either over- or under-estimated for all regions, simultaneously. Further note that, for this set of tests, the B-RL and C-RL

agents are not included for control since they are not impacted by the estimation errors of the critical accumulations. For a fair comparison, the MPC also assumes a mismatch of regional production between the predictive model and plant. Concretely, negative (positive) estimation errors indicate that the critical accumulations of the MFDs in the prediction model under- (over-) represent those of the MFDs in the plant. Note that, different from (*34*) where the MFDs are completely scaled up or down in the prediction model, here the MFDs are stretched to the left or right to simulate the estimation errors (see Fig. 11(a)). In this way, the MPC perceives the same maximum trip completion rate but different critical accumulations between the prediction model and plant, which is more similar to the estimation errors the DCK agents are subject to. In contrast, scaling the entire MFDs up or down would artificially cause worse performances for the MPC and unfairly highlight the proposed method.

Fig. 11(b) provides the control gains of different methods under each level of estimation error. As can be seen, both DCK agents can always outperform the MPC despite inaccurate critical accumulation information. This is remarkable since it shows that the effectiveness of the DCK is not contingent upon precise estimations of the critical accumulations and the agent can be applied for control regardless. The *B-RL+DCK* agent suffers from some performance degradation when the critical accumulation is significantly under-estimated (i.e., $\leq -10\%$). This is expected since considerably more accumulation values will be misclassified into the congested state, which then leads the agent to impose unnecessarily strict limitations on the transfer flows, i.e., excessive u_{min} values will be selected. Fortunately, the agent can realize stably higher CTC values than the MPC with mild under-estimation or varying levels of over-estimation errors as the agent is not too restrictive on the transfer flows. From a practical standpoint, these results suggest that the DCK is effective even under a wide range of estimation errors of the critical accumulations. However, for its best utilization, severe underestimation of the regional production should be avoided.

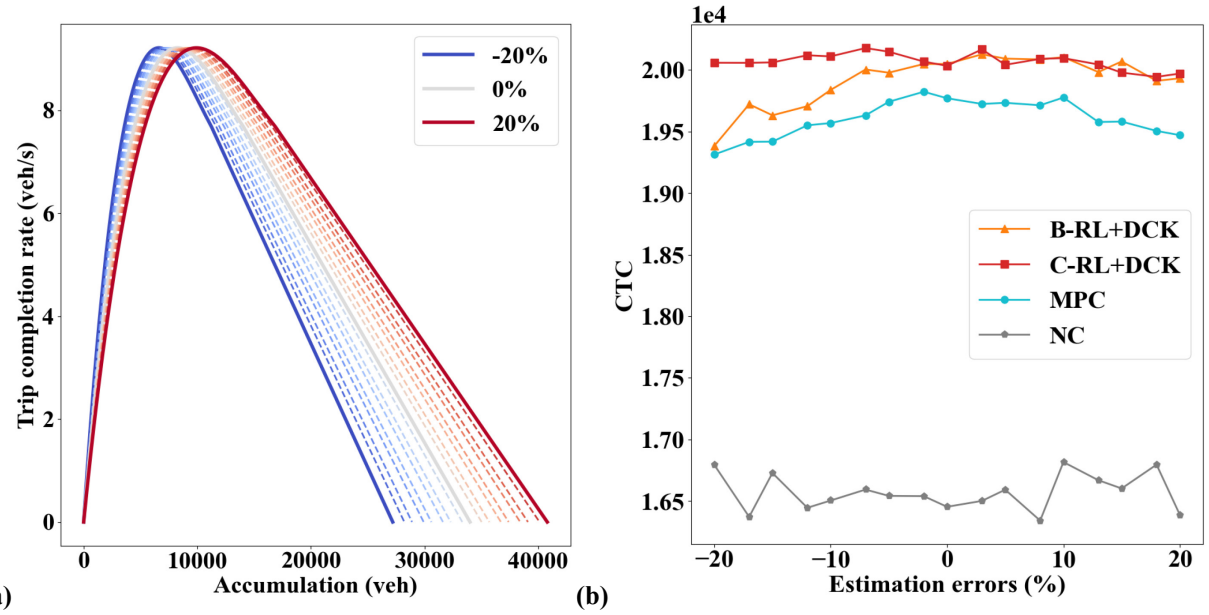

Fig. 11. (a) Illustration of estimation errors for the MPC; (b) Control gains with different estimation errors.

In summary, the robustness tests conducted in this section show that the DCK agents are resilient to various types of environment uncertainties. This is critical for real-world applications where uncertainties can be ubiquitous in sensors and network modeling techniques.

Three-Region Perimeter Control

To demonstrate the DCK effectiveness in a more general and complex setting, the three-region perimeter metering control problem is examined in this section. The experiment setup is first introduced, with extra details for the different control methods. The control results are then compared, which help illustrate the potential of the DCKs in larger perimeter control problems.

Experiment Setup

The three-region network (i.e., $N = 3$) considered is schematically shown in Fig. 12(a). The MFDs for R_1 and R_2 are separately scaled down by 20% and 10% from that of R_3 such that R_1 simulates a city center with the smallest MFD. The critical accumulations are $n_{1c} = 6{,}593$ veh, $n_{2c} = 7{,}417$ veh, $n_{3c} = 8{,}241$ veh. Like in two-region perimeter control, the traffic demands adopted here mimics a morning peak with higher inflows to R_1; see Fig. 12(b) where q_{21}, q_{31} are much higher than q_{12}, q_{13}. Note that, for more realistic simulations, the MFDs and traffic demands are subject to uncertainties as denoted by Eqs. (5)-(6) with $\alpha = \sigma = 0.2$. The initial accumulations are $n_1(0) = 8{,}000$ veh, $n_2(0) = n_3(0) = 6{,}500$ veh, where the city center is moderately congested while the outer regions are uncongested but exhibit different levels of vehicle presence.

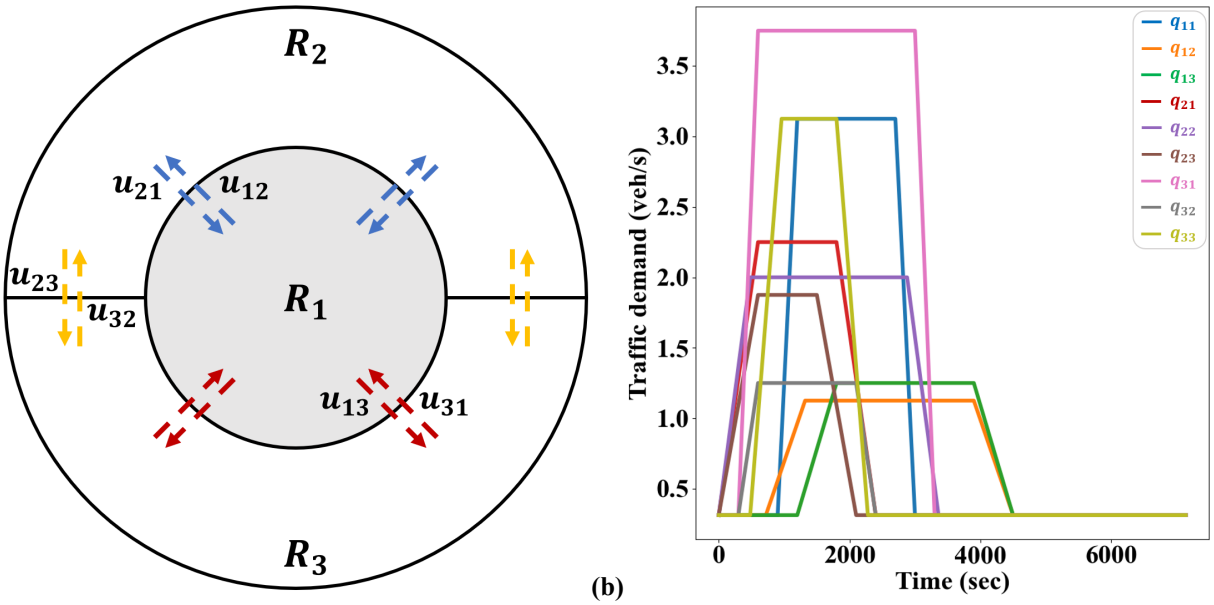

Fig. 12. (a) a three-region urban network; (b) estimated traffic demand profile.

For this problem, the state space consists of 3 regional accumulations $n_i(t)$ and 9 estimated traffic demands $q_{ij}(t)$, while the action space includes values for the six perimeter controllers $u_{ij}, i, j = 1,2,3, i \neq j$. Thus, both the B-RL and C-RL agents are adapted here to cope with expanded state and action spaces, i.e., the actor and critic networks for C-RL, as well as the Q-network for B-RL, are modified. However, the learning algorithms and the distributed learning architecture remain unchanged. The DCK with the above-derived representations is integrated with the two agents for control. Further, state spaces of the DCK agents are appended with the congestion information to "warm-start" the learning processes utilizing the default actions. All Deep-RL agents are compared with the MPC method, which is implemented according to the perimeter control-only scheme in (*35*) with a prediction horizon of 40 and a control horizon of 2. Note that, selection of the control horizon is consistent with prior works (*34*, *35*) and takes into consideration the extra problem complexity compared with two-region control. The control period is extended to 2 hours here, so the prediction horizon is extended as well.

Control Results

Fig. 13 presents the performance curves of different control methods for the three-region problem. As can be seen, the B-RL fails to learn effectively and underperforms the NC method throughout its learning course. While undesirable, this is not entirely unexpected due to difficulties in exploration. Concretely, the B-RL has a 64-dimension action space (two options for each of the six controllers), and exploration in such a high-dimensional space is not conceivably tractable, which thus leads to the agent's failure in learning. In contrast, the action space of the C-RL is only 6-dimensional, with one for each perimeter controller. Therefore, the C-RL can learn to control and realize relatively high network throughput at the end of learning. Comparing the curves between the original and DCK agents, one may notice that the DCK can significantly improve the learning and control performances of both agents, although the DCK is proposed only for a pair of neighboring regions. This indicates that even limited or imperfect knowledge can be readily and effectively applied to enhance the learning abilities of pure Deep-RL agents, a point also demonstrated in (*63*). Notably, the improved performances of the B-RL agent are made possible since it mainly explores around the most rewarding part of the action space with guidance provided by the DCK, thus greatly speeding the learning process. As such, the DCK can improve the scalability of the agents to larger problems by reducing the action space that needs to be explored. Motivated by this, the DCK agents might be potentially used as a sub-component for a larger-scale perimeter controller designed for city-level traffic management, which is considered as future extensions to the present work. Finally, notice that both DCK agents can often realize higher CTC values than the MPC as training proceeds. This is reasonable as the MPC is a model-based scheme that formulates a non-convex optimization program at each time step. For networks with increased number of regions, the size of the formulated program increases dramatically; for example, the three-region problem has 6 control variables whereas the two-region problem has only 2. Consequently, the expanding solution space, coupled with the non-convex nature of the formulated program as well as the high level of environment uncertainty, renders it increasingly difficult for the MPC to derive promising control policies, which explains its slight underperformance to the DCK agents.

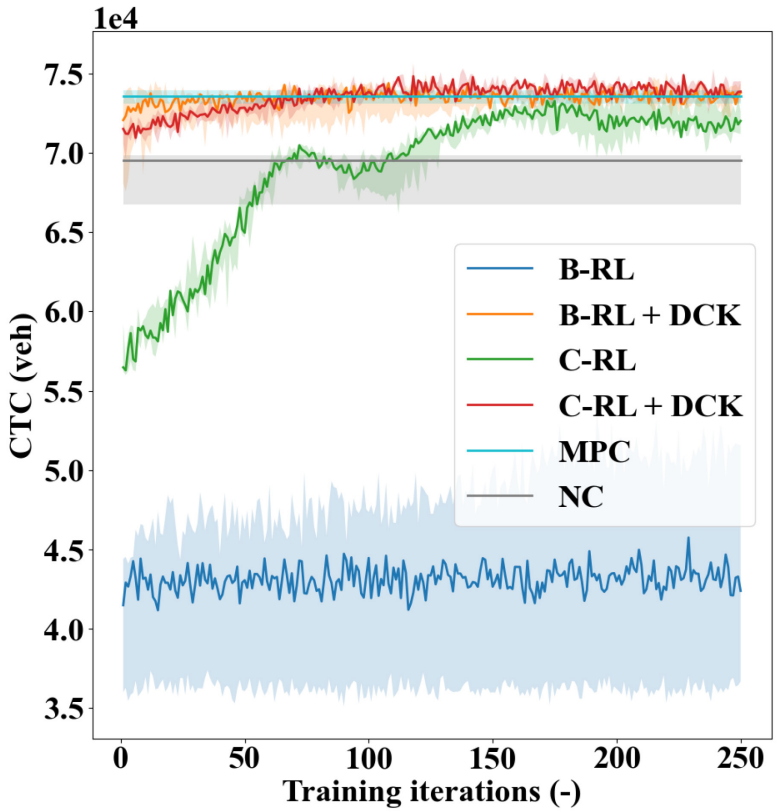

Fig. 13. Performance curves of the six control methods for three-region control.

The control outcomes of different methods in terms of the accumulation plots are provided in Fig. 14, where the critical accumulations are also included. A few observations can be made. First, under the NC policy, only region 1 exhibits notable congestion, which is reasonable since the experiment scenario simulates a morning peak with high inflows into the city center. The B-RL agent, however, produces a policy that leads to severe congestion both in region 1 and 3, which explains the poor throughput it realizes (i.e., with CTC values even lower than the NC). In comparison, the other methods can effectively mitigate the congestion in region 1, indicating fulfillment of the perimeter control objective, i.e., to protect regions from over-congestion by distributing the vehicles across the network. To this end, notice that these methods distribute vehicles in different manners. In particular, the C-RL agent reduces but not eliminates the congestion in the city center, while keeping both outer regions roughly uncongested. This scheme has some benefits but also some disadvantages, most notably the diminished throughput value due to congestion present in a region loaded with destinations, as opposed to the optimal AB strategy in (*5*). Nevertheless, the B-RL+DCK and C-RL+DCK agents respectively trade off efficiencies in region 2 and 3 for lessened congestion in the city center, whereas the MPC distributes vehicles throughout the network in a relatively balanced fashion. Therefore, these three methods can realize higher throughputs since the most destination-loaded region (i.e., the city center) has accumulations near or lower than the critical value. In addition, the similarities of the accumulation plots between the MPC and DCK agents imply that the DCK is indeed effective, even in larger problems.

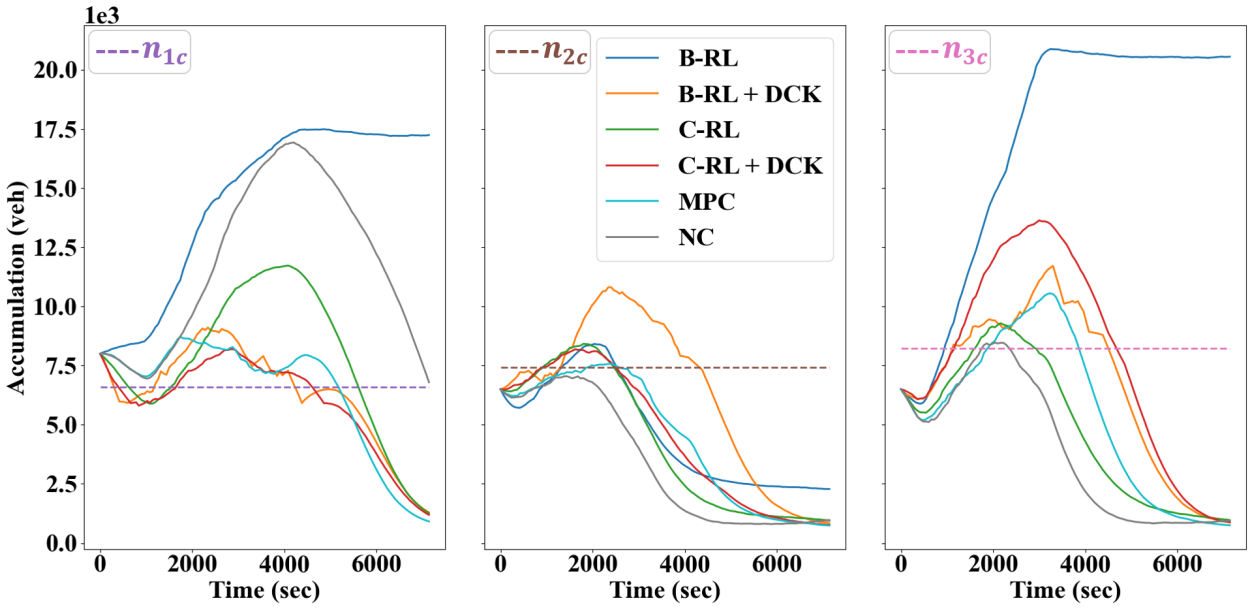

Fig. 14. Evolutions of accumulations by different control methods.

CONCLUDING REMARKS

This paper studies the two- and three-region perimeter control problems. A novel Deep-RL agent building upon the Bang-Bang policy is devised and utilized for control, together with the C-RL agent in a recent publication. The two agents exhibit initial underperformances to the no control method due to their random exploration of the entire action space. Hence, two types of DCK based on expert-level understanding of regional congestion dynamics are presented and integrated with the agents to improve their learning and control performances. Concretely, the DCK specifies the most fruitful part of the action space for the agents to explore by providing them with a set of sensible default actions at each action-taking step. A series of explorative experiments have been conducted to derive suitable representations for the DCK. Extensive numerical experiment results have shown the DCK can: (a) improve the learning and control performances for the agents; (b) improve the agents' resilience against various types of environment uncertainties (i.e., measurement noise of regional accumulations, estimation errors of the critical accumulations); (c) mitigate the scalability issue for the agents. Overall, the proposed DCK agents have been shown capable of achieving superior learning and control performances while in the meantime requiring little information about the environment. These results are promising for real-world applications of Deep-RL based regional control policies. Specifically, they suggest that Deep-RL agents could be integrated with DCK in a way that will not cause worse performances than the status quo (i.e., no control) in the course of learning, which would otherwise be grounds for removing the Deep-RL agents before they could fully learn about the environment.

Note that, while the comparability between the MPC and DCK agents is the main highlight of the present paper, it is not atypical for the latter to outperform the former, particularly in scenarios with high uncertainties and problem complexity; see Fig. 10-11 and 13. As touched upon previously, the MPC method is susceptible to the mismatch between the MFDs prediction model and plant, as prevalent in problems with high uncertainties. Also, in its open-loop problem, obtaining the global optimum cannot be guaranteed for the formulated high-dimensional non-convex program. These complications help explain the slight under-performance of the MPC to the proposed DCK agents and also justify the need to develop model-free data-driven control schemes. In this regard, note that it is computationally intensive for the MPC to solve the optimization program at every step, while the time needed to apply the pre-trained Deep-RL agents is nearly negligible. Though training the agents might take up an extended period, this could be done in a simulation platform offline. Overall, compared with the state-of-the-art MPC method, the proposed DCK agents have

been shown capable of achieving superior performances in large perimeter control problems with high uncertainties while in the meantime exhibiting prominent implementational efficiency at the time of application.

Future works could include developing DCK without using the critical accumulations to set the agents free from the estimation errors and to maintain the model-free property. Also, only two types of DCK are examined in this paper. It would be beneficial to research about other types of DCK, e.g., the one that is general enough to encompass all perimeter control problems. Further, it is a research priority to evaluate the proposed methods in a microsimulation platform, as opposed to the numerical simulations conducted herein. This would also establish grounds for possible real-world applications of the proposed method.

ACKNOWLEDGEMENTS

This research was supported by NSF Grant CMMI-1749200.

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. 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.

3. Mahmassani, H., J. C. Williams, and R. Herman. Performance of Urban Traffic Networks. 1987.

4. Smeed, R. J. The Road Capacity of City Centers. *Traffic Engineering & Control*, Vol. 9, No. 7, 1967, pp. 455–458.

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. 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.

7. 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.

8. 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.

9. 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.

10. Fu, H., Y. Wang, X. Tang, N. Zheng, and N. Geroliminis. Empirical Analysis of Large-Scale Multimodal Traffic with Multi-Sensor Data. *Transportation Research Part C: Emerging Technologies*, Vol. 118, 2020, p. 102725. https://doi.org/10.1016/j.trc.2020.102725.

11. Huang, C., N. Zheng, and J. Zhang. Investigation of Bimodal Macroscopic Fundamental Diagrams in Large-Scale Urban Networks: Empirical Study with GPS Data for Shenzhen City. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2673, No. 6, 2019, pp. 114–128. https://doi.org/10.1177/0361198119843472.

12. Paipuri, M., Y. Xu, M. C. González, and L. Leclercq. Estimating MFDs, Trip Lengths and Path Flow Distributions in a Multi-Region Setting Using Mobile Phone Data. *Transportation Research Part C: Emerging Technologies*, Vol. 118, 2020, p. 102709. https://doi.org/10.1016/j.trc.2020.102709.

13. Geroliminis, N., and J. Sun. Properties of a Well-Defined Macroscopic Fundamental Diagram for Urban Traffic. *Transportation Research Part B: Methodological*, Vol. 45, No. 3, 2011, pp. 605–617. https://doi.org/10.1016/j.trb.2010.11.004.

14. 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.

15. 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.

16. 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.

17. Gayah, V. V, X. S. Gao, and A. S. Nagle. On the Impacts of Locally Adaptive Signal Control on Urban Network Stability and the Macroscopic Fundamental Diagram. *Transportation Research Part B: Methodological*, Vol. 70, 2014, pp. 255–268.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. Su, Z. C., A. H. F. Chow, N. Zheng, Y. P. Huang, E. M. Liang, and R. X. Zhong. Neuro-Dynamic Programming for Optimal Control of Macroscopic Fundamental Diagram Systems. *Transportation Research Part C: Emerging Technologies*, Vol. 116, 2020, p. 102628. https://doi.org/10.1016/j.trc.2020.102628.

27. Ren, Y., Z. Hou, and T. Lei. Two-Region Macroscopic Traffic Network Perimeter Control via Model Free Adaptive Control Based Strategy. No. 2018-January, 2018, pp. 899–904.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. Lei, T., Z. Hou, and Y. Ren. Data-Driven Model Free Adaptive Perimeter Control for Multi-Region Urban Traffic Networks With Route Choice. *IEEE Transactions on Intelligent Transportation Systems*, 2019, pp. 1–12. https://doi.org/10.1109/tits.2019.2921381.

34. 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.

35. Sirmatel, I. I., and N. Geroliminis. Economic Model Predictive Control of Large-Scale Urban Road Networks via Perimeter Control and Regional Route Guidance. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 4, 2018, pp. 1112–1121. https://doi.org/10.1109/TITS.2017.2716541.

36. 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.

37. 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.

38. Yildirimoglu, M., I. I. Sirmatel, and N. Geroliminis. Hierarchical Control of Heterogeneous Large-Scale Urban Road Networks via Path Assignment and Regional Route Guidance. *Transportation Research Part B: Methodological*, Vol. 118, 2018, pp. 106–123. https://doi.org/10.1016/j.trb.2018.10.007.

39. Prabhu, S., and K. George. Performance Improvement in MPC with Time-Varying Horizon via Switching. 2014.

40. Schrangl, P., T. Ohtsuka, and L. Del Re. Parameter Sensitivity Reduction of Nonlinear Model Predictive Control for Discrete-Time Systems. No. 2018-January, 2018, pp. 2131–2136.

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. 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.

43. Zhou, D., and V. V. Gayah. Model-Free Perimeter Metering Control for Two-Region Urban Networks Using Deep Reinforcement Learning. *Transportation Research Part C: Emerging Technologies*, Vol. 124, 2021, p. 102949.

44. Aalipour, A., H. Kebriaei, and M. Ramezani. Analytical Optimal Solution of Perimeter Traffic Flow Control Based on MFD Dynamics: A Pontryagin's Maximum Principle Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 9, 2019, pp. 3224–3234. https://doi.org/10.1109/TITS.2018.2873104.

45. Saeedmanesh, M., and N. Geroliminis. Clustering of Heterogeneous Networks with Directional Flows Based on "Snake" Similarities. *Transportation Research Part B: Methodological*, Vol. 91, 2016, pp. 250–269. https://doi.org/10.1016/j.trb.2016.05.008.

46. Ji, Y., and N. Geroliminis. On the Spatial Partitioning of Urban Transportation Networks. *Transportation Research Part B: Methodological*, Vol. 46, No. 10, 2012, pp. 1639–1656. https://doi.org/10.1016/j.trb.2012.08.005.

47. Saeedmanesh, M., and N. Geroliminis. Dynamic Clustering and Propagation of Congestion in Heterogeneously Congested Urban Traffic Networks. *Transportation Research Part B: Methodological*, Vol. 105, 2017, pp. 193–211.

48. Yildirimoglu, M., M. Ramezani, and N. Geroliminis. Equilibrium Analysis and Route Guidance in Large-Scale Networks with MFD Dynamics. *Transportation Research Part C: Emerging Technologies*, Vol. 59, 2015, pp. 404–420. https://doi.org/10.1016/j.trc.2015.05.009.

49. Sirmatel, I. I., and N. Geroliminis. Stabilization of City-Scale Road Traffic Networks via Macroscopic Fundamental Diagram-Based Model Predictive Perimeter Control. *Control Engineering Practice*, Vol. 109, 2021, p. 104750. https://doi.org/10.1016/j.conengprac.2021.104750.

50. Kouvelas, A., M. Saeedmanesh, and N. Geroliminis. A Linear Formulation for Model Predictive Perimeter Traffic Control in Cities. *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 8543–8548. https://doi.org/10.1016/j.ifacol.2017.08.1411.

51. Li, Y., M. Yildirimoglu, and M. Ramezani. Robust Perimeter Control with Cordon Queues and Heterogeneous Transfer Flows. *Transportation Research Part C: Emerging Technologies*, Vol. 126, 2021, p. 103043. https://doi.org/10.1016/j.trc.2021.103043.

52. Keyvan-Ekbatani, M., R. C. Carlson, V. L. Knoop, and M. Papageorgiou. Optimizing Distribution of Metered Traffic Flow in Perimeter Control: Queue and Delay Balancing Approaches. *Control Engineering Practice*, Vol. 110, 2021, p. 104762.

53. Abdulhai, B., R. Pringle, and G. J. Karakoulas. Reinforcement Learning for True Adaptive Traffic Signal Control. *Journal of Transportation Engineering*, Vol. 129, No. 3, 2003, pp. 278–285. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:3(278).

54. 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.

55. Liang, X., X. Du, G. Wang, and Z. H. Fellow. *Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks*. 2018.

56. 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.

57. Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous Control with Deep Reinforcement Learning. 2016.

58. Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic Policy Gradient Algorithms. 2014.

59. 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.

60. Lin, L.-J. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. *Machine Learning*, Vol. 8, No. 3–4, 1992, pp. 293–321. https://doi.org/10.1007/bf00992699.

61. Horgan, D., J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver. Distributed Prioritized Experience Replay. 2018.

62. Gao, X. (Shirley), and V. V. Gayah. An Analytical Framework to Model Uncertainty in Urban Network Dynamics Using Macroscopic Fundamental Diagrams. *Transportation Research Part B: Methodological*, Vol. 117, 2018, pp. 660–675. https://doi.org/10.1016/j.trb.2017.08.015.

63. Zhang, P., J. Hao, W. Wang, H. Tang, Y. Ma, Y. Duan, and Y. Zheng. KoGuN: Accelerating Deep Reinforcement Learning via Integrating Human Suboptimal Knowledge. 2020, pp. 2291–2297.