TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–17


Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/03611981241230313
journals.sagepub.com/home/trr
Sage

Evaluating the Effectiveness and Transferability of a Data-Driven Two-Region Perimeter Control Method Using Microsimulation

Dongqin Zhou[1] and Vikash V. Gayah[1]

Abstract

Regional perimeter control based on the existence of macroscopic fundamental diagrams has been widely studied as an effective tool to regulate traffic and prevent oversaturation in dense urban areas. Significant research efforts have been performed concerning the modeling aspects of perimeter control. More recently, data-driven techniques for perimeter control have shown remarkable promise; however, few studies have examined the transferability of these techniques. While it is surely of the highest priority to devise effective perimeter control methods, the ability of such methods to transfer the learned knowledge and quickly adapt control policies to a new setting is critical, particularly in real-life situations where training a method from scratch is intractable. This work seeks to bridge this research gap by comprehensively examining the effectiveness and transferability of a reinforcement-learning-based perimeter control method for a two-region urban network in a microsimulation setting. The results suggest: 1) the presented data-driven method demonstrates promising control effectiveness in comparison with no perimeter control and an extended greedy controller and 2) the method can readily transfer its learned knowledge and adapt its control policy with newly collected data to simulation settings with different traffic demands, driving behaviors, or both.

Keywords

operations, advanced technology, microscopic traffic simulation, traffic management and control, macroscopic traffic models, traffic control, traffic flow

Urban traffic signal control has been a heated research topic in the transportation community owing to its potential to alleviate congestion and reduce accidents. Classical signal control systems such as SCATS and SCOOT make the utilization of existing transportation infrastructures more efficient (*1*, *2*). However, these systems are microscopic and localized ones that concentrate on intersection- or link-level performances yet disregard network-level effects such as congestion propagation. Further, their applicability and effectiveness may be in question under oversaturated traffic conditions where severe queue spillbacks occur. Historically, there have been continued efforts on aggregate modeling of traffic dynamics to facilitate the development of network-level traffic control schemes, with the notion of macroscopic fundamental diagram (MFD) receiving extensive research interest over the past 15 years. Initially proposed in (*3*) and recently verified with analytical and empirical evidence (*4, 5*), MFD describes a well-defined relationship between network usage (e.g., vehicle accumulation) and production (e.g., space-mean traffic flow or trip completion rate). The existence of MFD has been observed in homogeneously loaded traffic networks with low spatial variation of traffic flows (*5*, *6*). In the presence of heterogeneous vehicle distributions, undesirable phenomena such as instability, hysteresis, and bifurcation may arise (*7–10*). In such scenarios, network partitioning methods can be applied to divide a heterogeneous network into several smaller

[1]Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA

Corresponding Author:
Dongqin Zhou, dongqin.zhou@psu.edu

regions so as to maintain the feasibility of MFD-based aggregate modeling of traffic dynamics (*11*, *12*).

Ever since its conceptualization, MFD has served as a theoretical foundation in the construction of numerous regional-level traffic control schemes. Among these, perimeter control, which involves the regulation of transfer flows between neighboring regions, is perhaps the most intensively studied. Adopting a gating concept, perimeter control aims to regulate the regional accumulations to predefined critical levels that are associated with the maximum productions. This control scheme is particularly helpful in scenarios with strong directional traffic demands into a protected region where the resultant congestion cannot be mitigated by intra-regional traffic control alone. Numerous works have been presented in the literature, with perimeter control applications spanning single-region, two-region, and multi-region urban networks (*4*, *13–19*). Different aspects of perimeter control have also been examined, for example the integration with route guidance, robust control, and boundary queue dynamics (*18*, *20–24*).

Over the years, various techniques have been presented for perimeter control problems. These range from classical proportional–integral regulator to more advanced model predictive control (MPC) (*15*, *18*, *21*, *25*). Recent times have witnessed an increasing trend to develop data-driven methods for perimeter control, which are particularly helpful as a network's traffic dynamics or MFD functions are often unknown or hard to calibrate accurately. Even more so, the modeling of network traffic dynamics may need to be adjusted continuously as it is prone to exogeneous disturbances (e.g., vehicle rerouting and demand stochasticity). These modeling difficulties thus highlight the prospect of data-driven perimeter control methods, in contrast to the model-based counterparts. On this note, the modeling difficulties also highlight the need to evaluate perimeter control methods in a more realistic environment that does not rely on the explicit modeling of traffic dynamics (e.g., microsimulation rather than numerical simulation) as, otherwise, comparisons using inaccurate models may not truthfully reflect the real efficacy of the methods.

Despite still being in the early stage, the development of data-driven methods for perimeter control has seen some notable research works; for example, data-driven adaptive control methods, adaptive dynamic programming methods, and reinforcement learning (RL)-based approaches (*26–33*). The current paper follows the lines of these works to present an RL-based perimeter controller and evaluate its effectiveness and transferability using microsimulation. Note, most data-driven methods are evaluated using numerical experiments with explicit models of network macroscopic traffic dynamics. However, these models are merely estimates; more realistic evaluations and impartial comparisons can only be established with microsimulation. To the best knowledge of the authors, (*29, 30*) are the only works that have presented and evaluated data-driven perimeter control methods using microsimulation. In (*29*), numerical experiments and microsimulations were adopted for a comprehensive validation, while in (*30*) the perimeter controller is evaluated in combination with the max pressure method (*34*). However, only a single-region network was simulated there, where boundary queue impacts were largely overlooked. Importantly, neither work considered the transferability aspects of their methods.

While it is surely of the highest priority to devise an effective perimeter control method, the ability of the method to transfer to unencountered traffic conditions is also critical, especially in real-life scenarios where training the method from scratch may be costly or even infeasible. Moreover, in practical applications of perimeter control, the scenarios where the data-driven methods are to be applied (i.e., reality) may differ from where they are initially trained (i.e., simulation). In such cases, the ability of the methods to transfer the learned knowledge and even to continue their learning trajectories is crucial, as this would significantly reduce the training time thus facilitating fast application on the new scenarios. In this regard, note that, while transferability has been considered in (*31*, *33*), only numerical experiments were adopted, and the ability of the proposed agents to keep learning in another setting with continuous data feed is not examined (*31*, *33*). Further, the current work focuses on examining the effectiveness and transferability of a RL-based two-region perimeter controller in microsimulation and seeks to bridge several research gaps previously outlined, while acknowledging that perimeter control is merely part of the urban traffic control framework (*31–33*). Combining this regional-level control with intra-regional traffic signal control will potentially increase the control benefits and lead to a more complete paradigm of urban traffic control, as previously demonstrated in, but this exceeds the scope of the present work (*30*, *35*, *36*). The significance of this work is to show that the RL controller, when pretrained in one setting, can transfer its knowledge and more quickly adapt its action-taking policy (compared to training from scratch) with newly collected data in a different setting with demand patterns, driving behaviors, or both, that are more indicative of reality.

The remainder of the paper is structured as follows. The next section provides the specification of the simulated two-region network. The methodology is then explained, followed by the simulation results on the evaluation of effectiveness and transferability. Concluding remarks are provided in the last section.

Two-Region Urban Network Set-Up

In this work, a two-region urban network is simulated using the Eclipse SUMO software (*37*), where a larger periphery encompasses a smaller city center (see Figure 1). The periphery shaded in blue is denoted as Region 1 while the city center in gray is denoted as Region 2. The two regions are assumed to be homogeneous and connected by two-directional linking roads where perimeter control can be enforced (see (*28*) for a similar structure). Figure 1 also presents the detailed layouts of three types of intersections in the network: four-leg (circled in orange), T-shaped (circled in blue), and the perimeter control intersections (circled in green). Each street in the network assumes a length of 500 m with three lanes in each travel direction. The free flow speed of each lane is set to 50 km/h while the saturation flow is 1,800 vehicles per hour per lane.

All intersections in the network are signalized, where the non-perimeter control intersections adopt a fixed multiphase signal plan with a shared cycle length of 90 s. In contrast, the 16 perimeter control intersections assume a common cycle length of 30 s so that they can adapt faster to the prevailing traffic. All left-turn movements are treated as protected, as permitted movements form long queues and were observed to be a source of inhomogeneity. No offset is assumed, as it is shown to be inconsequential to the network-level performances in grid networks (*38*). The simulation step is set to 1 s.

Origin and destination locations are evenly distributed within each region. To simulate scenarios where perimeter control is the most helpful—that is, to protect destination-loaded regions from over-saturation—strong directional demand from the periphery to the city center is assumed; see Figure 2 for the baseline demand profile where strong inbound traffic flows last for 60 min followed by a recovery period of 30 min. Each of the total demands (e.g., from Region 1 to 2) is evenly assigned to all associated origin-destination pairs. Note that, while the traffic demands in Figure 2 appear to be constant, the realized traffic demands will exhibit variability during each simulation instance; for example, the exact times when vehicles are inserted into the network, the initial routes of the vehicles, and/or the vehicle speeds may differ in each simulation instance depending on the random number generation process. For this reason, multiple random seeds were used to enhance realism for the traffic demands. Moreover, different demand profiles will be adopted in subsequent sections. The simulated vehicles are initially routed using the stochastic C-logit route choice model (*39*). A subset of the vehicles (60%) were assumed to be able to adaptively reroute themselves based on prevailing traffic to mimic more realistic driving patterns (which, other than rerouting, also include car-following and lane-changing behaviors). This adaptive rerouting has been shown to be helpful to network-wide operational performances (*7*, *8*). In this work, it happens at regular intervals of 3 min.

Methodology

This section first presents designs of a comparative method, that is, an improved greedy controller (I-GC).

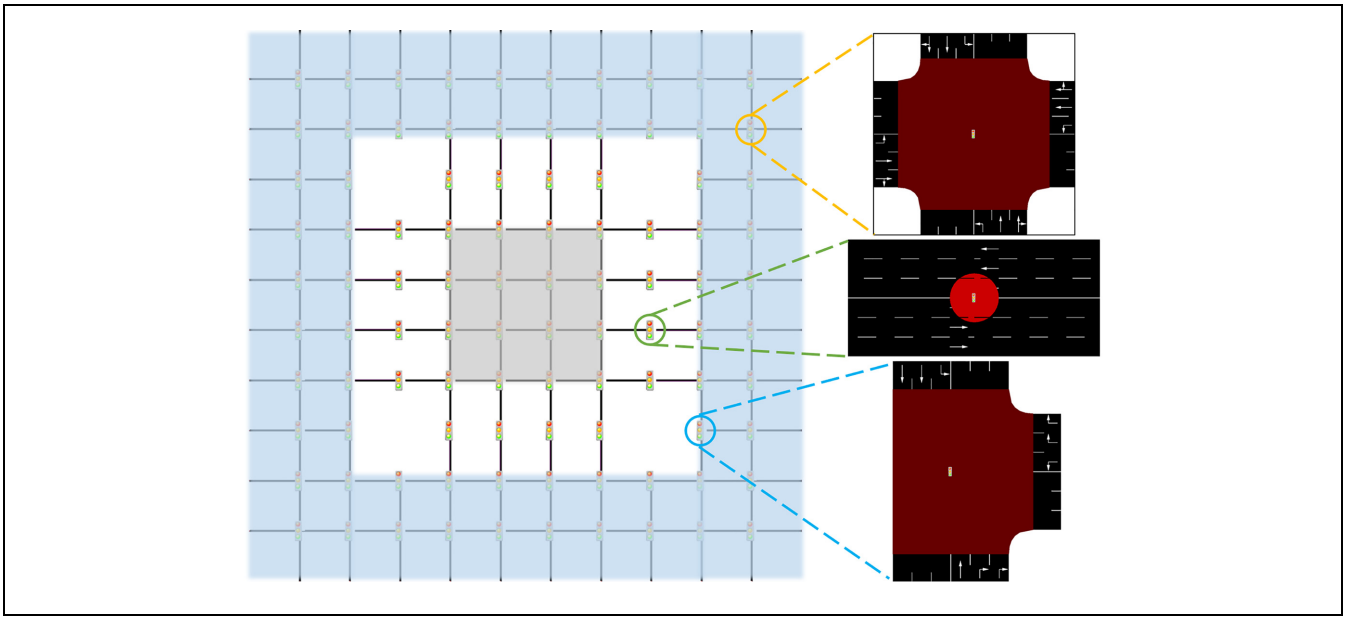

Figure 1. The simulated two-region urban network.
Note: blue shading = periphery (Region 1); gray shading = city center (Region 2); orange circle = four-leg intersection; blue circle = T-shaped intersection; green circle = perimeter control intersection.

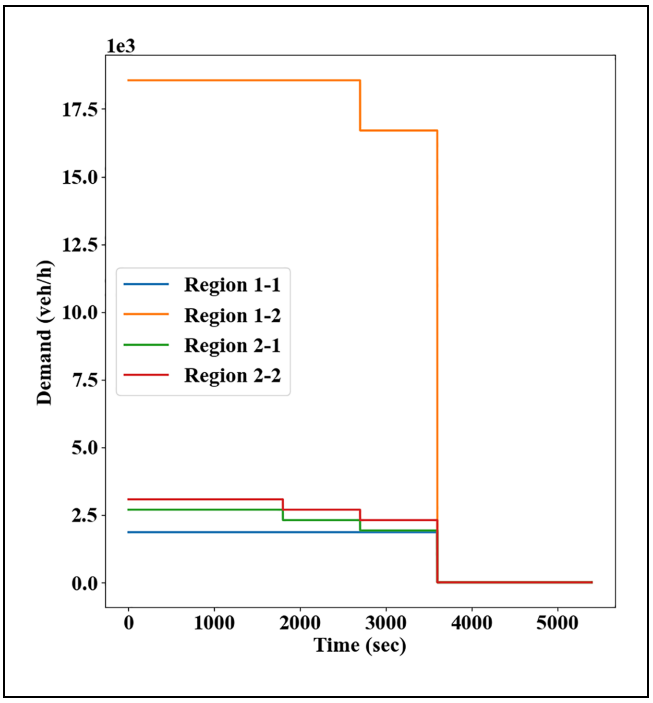

Figure 2. The baseline traffic demand profile.

Adopting similar design ideas, the RL-based two-region perimeter controller is explained. Lastly, transfer learning—a technique that enables the transferring of learned knowledge and policies to a new setting—is discussed. Note, in this work the MPC method is not applied for comparison, as consistent with existing works that evaluate data-driven perimeter control methods using microsimulation (*29*, *30*). The reasons are multifold. For one, the MPC method has tremendous data requirement, such as detailed traffic demand information throughout the whole simulation. However, such information is specific to the environment (microsimulation or reality) and often not made available to the controller beforehand. For another, applying MPC necessitates the estimation of the MFD functions so as to describe the traffic evolutions using dynamic equations in the prediction model. Yet, such estimation (and therefore the resultant prediction model) is prone to significant errors, which would make the MPC application particularly challenging (*8*–*10*). In contrast, I-GC can serve as an effective comparative baseline for perimeter control in a microsimulation environment, as will be explained shortly.

Improved Greedy Controller (I-GC)

Greedy control, a two-region extension of the bang-bang policy, seeks to protect the more congested region by minimizing the transfer flow into it, and it has often been adopted as a comparative baseline (*4*). However, despite its relatively wide usage involved in numerical experiments, its effectiveness has not been examined in microsimulation environments (*15*, *40*). The greedy control policy considers two levels of congestion in the MFDs (with accumulations below or over the critical values) and could alternate its action abruptly around the critical values. Nevertheless, the regions are operating roughly around the maximum production level in the proximity of the critical values, and such abrupt alternation may cause irregularities in the traffic patterns that would disrupt the congestion distribution and evolution. While such impacts may be negligible in numerical simulations where vehicle dynamics are not modeled, they could greatly manifest themselves in a microsimulation environment and cause local pockets of congestion that cannot recover.

Microsimulation experiments are used in this work; thus, the greedy control policy is extended with consideration of three levels of congestion in the MFDs: free flow, critical flow, and congestion flow. A similar categorization of the regional congestion levels can also be found in (*32*, *41*). Specifically, when a region operates in free flow, traffic flows into the region should not be metered, and green times at the perimeter control intersections should be set at the maximum value (g_{max}). When the region operates in critical flow, it is prone to worsened congestion and even instability once the accumulation becomes sufficiently high (*7*). As such, the inflow should be metered. However, if the inflows are strictly metered, significant queues will form around the perimeter control intersections and may even spill back to the upstream locations. Thus, as a middle ground, green times should be set at a small value (g_{mid}) that is close to the minimum value (g_{min}). When the region operates in congestion, the regional productivity becomes fairly low and it may be gridlocked should the congestion not be restored soon. In this case, vehicles that enter the region may not be able to reach their destinations in a timely manner and may even impede completion of trips that are already within the region. Therefore, the strictest metering should be enforced to help restore the regional operations, and the green times should be set at g_{min}. Integrating these intuitions into a two-region urban network system, the I-GC policy is summarized in Table 1, where the green times at the perimeter control intersections for traffic flows from Region 1(2) to Region 2(1) are denoted as g_{12} (g_{21}); the accumulations and the jam values of the two regions are $n_i, i = 1, 2$ and $n_{i,jam}, i = 1, 2$; and $n_{i,c1}, n_{i,c2}, i = 1, 2$ represents two cutoff points used to categorize the congestion levels for each region.

A few remarks are provided here on the I-GC policy. First, the cutoff points generally adopt values higher

Table 1. Improved Greedy Control (I-GC) Policy

$(g_{12}(t), g_{21}(t))$	$n_1 \in [0, n_{1,c1})$	$n_1 \in [n_{1,c1}, n_{1,c2}]$	$n_1 \in (n_{1,c2}, n_{1,jam}]$
$n_2 \in [0, n_{2,c1})$	(g_{max}, g_{max})	(g_{max}, g_{mid})	(g_{max}, g_{min})
$n_2 \in [n_{2,c1}, n_{2,c2}]$	(g_{mid}, g_{max})	(g_{mid}, g_{mid})	(g_{mid}, g_{min})
$n_2 \in (n_{2,c2}, n_{2,jam}]$	(g_{min}, g_{max})	(g_{min}, g_{mid})	na*

Note: na = not applicable. *This case is not considered, as perimeter control will be of little help.

than the critical accumulations to avoid excessively long queues at the perimeter control intersections that may be highly disruptive to the network operations. In such manner, the critical flow range $[n_{i,c1}, n_{i,c2}]$ spans a little over the critical accumulations, whereas the congestion flow range $[n_{i,c2}, n_{i,jam}]$ indicates severe regional congestion. Second, the traffic demands are assumed to be feasible, that is, the resultant congestion can be managed by traffic control. As such, the scenario where both regions are severely congested is not considered since, in that case, perimeter control may be of little help. This is reflected by the empty cell ("na") in Table 1. Third, notice that the I-GC policy directly manipulates the green times at the perimeter control intersections, instead of deriving those from allowable ratios of transfer flows (*17*, *19*). This form of implementation is mainly considered to alleviate the data requirement during perimeter control application. Specifically, as implied in (*17, 19*), the detailed current traffic flows on each perimeter control intersection must be collected to minimize the differences between the actual transfer flows and ordered ones (*17*, *19*). However, in a realistic network, obtaining such detailed and large amount of information is not always tractable. Similar to the green time manipulations, recent works have considered setting the green ratios at the perimeter control intersections (*29*, *30*). Lastly, compared with greedy control, the I-GC policy is more flexible and can smooth the resulting traffic patterns by providing a control buffer between complete metering and accommodation of transfer flows. Concretely, the vanilla greedy control policy, without considering the congestion level of critical flow, would alternate the action between g_{min} and g_{max}. If g_{min} indicates complete metering (i.e., with a value of 0), then significant queues will form at the perimeter control intersections even if the protected region is marginally congested. As a result, the regional capacity will be under-utilized when the protected region becomes marginally uncongested. On the other hand, if g_{min} is greater than 0, then the regional congestion cannot be adequately metered if the region is already severely congested (as vehicles can still enter during the green time of g_{min}). The I-GC policy thus mitigates all these drawbacks. In addition, notice that at the core of the I-GC policy lies the bang-bang form of actions (as adopted by vanilla greedy control), which has been shown optimal for perimeter control problems (*4*, *24*, *40*). The extensions serve to further improve its applicability in a microsimulation setting. This applicability, coupled with the eased data requirement, renders I-GC an effective comparative baseline for perimeter control in microsimulation environments.

Reinforcement Learning (RL)-Based Two-Region Perimeter Controller

To better compare with the I-GC policy, the RL controller utilizes the same actions; that is, it selects among $g_{min}, g_{mid}, g_{max}$ as the green times for both travel directions at the perimeter control intersections. As such, the action space is 9-dimensional (three possible choices for each travel direction), and faithful comparisons with I-GC can be established with this action space. The controller does not embed into its designs explicit information about the environment and aims to learn a reasonable acting policy from interactions with the environment. However, when initialized, the controller has a rather randomized policy. To conduct effective learning, it first needs to actively explore the environment based on the inputs it receives (i.e., the state) and a random action-taking process, as follows:

$$\pi(s_t) = \begin{cases} \underset{a_t}{arg\,max}\, Q(s_t, a_t; \theta_t), & probability\ 1-\epsilon \\ a\ random\ action, & probability\ \epsilon \end{cases} \quad (1)$$

where

s_t = the state at step t (which includes the vehicle accumulations, average speed and flow of both regions),

a_t = the action at step t (i.e., the green times at the perimeter control intersections),

$\pi(\cdot)$ = the policy that map a state to an action,

$Q(\cdot)$ = the long-term benefits of taking the action a_t at state s_t (also known as action value), and

ϵ = a user-defined parameter that controls the amount of exploration the controller performs in its learning process.

Clearly, larger values of ϵ indicate more exploration of the environment (more random actions) and therefore more variability in the realized control outcomes. To

reduce the opportunity loss during exploration, the ϵ value is decayed in the training process. Further, the action value predictions $Q(\cdot)$ are parameterized by θ_t, and in this work θ_t represents the weight and bias parameters of a neural network, as widely adopted in the deep-RL literature (*42–44*). Intuitively, Equation 1 suggests the RL controller takes an action with a blend of its own knowledge (the predicted action values) and a random exploration process to ensure that it can sufficiently explore the environment whilst learning a reasonable action-taking policy by learning the parameters θ_t.

The actions taken by the RL controller, which inform the green time settings at the perimeter control intersections, are then implemented in the microsimulation for a time step, which in this work has the same duration as the cycle length of the perimeter control intersections (i.e., 30 s). At the end of a time step, the microsimulation arrives at a new state and returns a reward to the controller as an evaluation for the action taken. The reward is characterized by the weighted sum of average flows in both regions, and a larger weight (0.7) is placed on the average flow of the inner region (i.e., Region 2, the simulated city center) since it is the region to protect from oversaturated traffic conditions. With this reward, the RL controller proceeds to take another action based on the new state. This action-taking process is executed sequentially until termination, that is, until the simulation ends in 90 min.

The learning goal of the RL controller is to accumulate as many rewards as possible in a simulation run. To balance the importance of rewards with respect to the time they are received, a discount factor is employed which decays the value of the rewards received at a delayed time. Thus, the learning goal amounts to maximizing the cumulative discounted reward in a simulation, as termed by return G and calculated by

$$G = \sum_{\tau=0}^{T} \gamma^{\tau} r_{\tau+1} \tag{2}$$

where
$\gamma \in [0, 1]$ = the discount factor,
$r_{\tau+1}$ = the reward for the action taken at time step τ, and
T = the total time steps of a simulation.

Note, the action value $Q(\cdot)$ presented above is an estimate of the return G starting at state s_t and action a_t. Therefore, when not exploring, the RL controller is attempting to maximize the estimated return by choose the action $\arg\max_{a_t} Q(s_t, a_t; \theta_t)$ (see Equation 1).

To carry out the learning process in a principled manner, the double DQN method is adopted (*43*). Specifically, after each visit to a state-action pair (s_t, a_t), the RL controller receives a reward r_{t+1} from the environment and constructs a learning target Y_t as follows:

$$Y_t = r_{t+1} + \gamma Q\left(s_{t+1}, \arg\max_{a} Q(s_{t+1}, a; \theta_t); \theta_t^-\right) \tag{3}$$

where
θ_t^- = parameters of the target network (a periodical copy of the parameters θ_t that is used to stabilize the learning process by providing relatively static learning targets).

The learning target embeds in the Bellman Equation for solving Markov decision processes and also decouples the action selection from evaluation to mitigate the overestimation of action values (*43*, *45*, *46*). Then, the RL controller updates its parameters toward the learning targets by minimizing the loss:

$$\mathcal{L} = \sum_{i=0}^{b} \left[Y_t^i - Q(s_t, a_t; \theta_t)\right]^2 \tag{4}$$

where
b = the size of a minibatch of transitions $<s_t, a_t, r_{t+1}, s_{t+1}>$ sampled from the replay buffer (*47*).
Note, utilizing a replay buffer improves sample efficiency as it allows reusing the transitions multiple times; also, it enhances training stability by reducing correlations between the transitions via random sampling (47).

In this work, the parameters θ_t to be optimized represent the weight and bias parameters of a neural network constructed as a two-layer 64-unit multilayer perceptron. To minimize the loss in Equation 4, the Adam optimizer is used to perform stochastic gradient descent (*48*). To estimate the parameters more accurately, a distributed learning structure is also integrated with the RL controller to collect an increased amount of transitions (*44*). The training process of the RL controller lasts for 100 iterations, and four simulations are run concurrently in each iteration to gather more training experiences. During the learning process, the ϵ value is decayed from 1 to 0.05 exponentially by 5% per iteration, while the learning rate decreases from 0.003 to 0.001. The discount factor is set to 0.95 to account for the potentially delayed effects of perimeter control. The replay buffer contains the past 10,000 transitions and in each iteration 1,000 transitions are sampled to update the parameters θ_t. Parameters of the target network θ_t^- are replaced with θ_t every five iterations.

Transfer Learning

Transferability is an important methodological aspect for a perimeter control method to have the potential to be applied in the real world. While training a perimeter control method from scratch can be conveniently done using numerical experiments or microsimulation, this is hardly feasible in real networks. Further, the training process can be rather time-consuming, even under microsimulation. As such, the ability of a method to transfer its learned knowledge and continue the learning course

becomes increasingly crucial. Though transferability of a perimeter control method has been investigated by (*31*, *33*), only numerical experiments were considered and the ability of the methods to continue learning with online data feed was neglected. This work thus bridges a few research gaps outlined there.

In the present work, the examination of transferability is enabled by transfer learning, a technique that helps speed up fast application of knowledge gained from one problem to another. Typically, transfer learning is applied on deep learning tasks (such as computer vision) where there is a lack of sufficient data (e.g., labeled images) on a new problem, in which case the general features learned from a relevant problem can be reused. In this work, instead of transferring the learned features, the mapping from state-action pairs to action values (i.e., $Q(s_t, a_t; \theta_t)$) is transferred to a new task with the learned parameters θ_t being reused (see similar concepts of value function transfer in (*49*, *50*). Specifically, the transferred setting (called the "target problem") has different traffic demands, driving behaviors, or both, than the source setting (as previously described in the network setup), while the definitions of states, actions, and rewards are not altered. As such, directly transferring the learned action value function is feasible. Further, RL controller is trained to conduct effective perimeter control based on the traffic conditions of both regions (as indicated by the state). While the detailed action-taking policy may not apply to the target problem, the principles underlying the policy learned from the source setting should resemble those for the target problem. Therefore, transferring the action value function could potentially provide a better starting point for the target problem. Moreover, in the target problem, continued streams of data (in the form of transitions $<s_t, a_t, r_{t+1}, s_{t+1}>$) can be collected and fed to the RL controller so that it can adapt the transferred value function quickly and thus produce improved perimeter control policies. Note, the pretrained RL controller has already internalized knowledge on how to act on the traffic conditions; therefore, less learning update and exploration will be needed for the controller during adaptation. The reader may also refer to (*51, 52*) for more information on transfer learning and its relevance to reinforcement learning.

This work considers only homogeneous transfer learning (*53*) applied to perimeter control; that is, the source and target problems share the same network settings such as intersection layouts and speed limit, while the traffic patterns (e.g., travel demands and driver rerouting behaviors) are allowed to differ. In the case of notable differences between the source and target problems (e.g., when there are numerous road closures in the target problem), negative transfer might occur, that is, the reused knowledge is not helpful to the target problem and ends up hampering the learning process rather than accelerating it. As such, transferring the agent might perform even worse than training the agent from scratch. On the other hand, homogeneous transfer learning allows the RL controller to adapt quickly to unseen traffic patterns where there may be travel demands, driving behaviors, or both, different from microsimulation and more representative of reality.

To sum up, in this work, transferability is evaluated by transferring the learned action value function of a pretrained RL controller to a target problem and adapting the function with continued data feed. With this type of intra-agent transfer (*51*), the RL controller could learn better initial control policies and improve the control performances more efficiently with reduced learning and exploration, compared with training the controller from scratch.

Microsimulation Experiments

In this section, the effectiveness and transferability of the RL controller is evaluated on a simulated two-region network. The experiment setups are first provided, followed by simulation results.

Experiment Setup

Initially, to investigate the regional productions (MFD), a strong traffic demand is used to fill up the network. The MFDs (reflected by regional trip completion rate versus accumulation relationships) are presented in Figure 3, where each symbol represents a random seed in the simulation. Note, trip completion refers to a trip ended within the region or transferred to the neighboring region; also, the term MFD is used interchangeably as the network exit function (*15*, *28*, *29*). Each point in Figure 3 represents average measurements of trip completion and vehicle accumulation within an interval of 180 s. As can be observed, both regions exhibit relatively low-scatter MFDs, and the regional trip completion rate peaks at accumulations around 5,000 vehicles and 1,000 vehicles for Region 1 and 2, respectively. The macroscopic traffic dynamics are not modeled as they are neither utilized in the microsimulation nor the control algorithm, so the MFD functions are not estimated. The critical accumulation information, however, provides guidance on how the comparative benchmarking methods should be implemented (but not needed for the RL method).

Two benchmarking methods are considered, that is, no control (NC) and the I-GC policy. In the case of NC, fixed time signal plans are applied, which simulates the status quo. I-GC is the baseline perimeter controller and used to show perimeter control can alleviate oversaturated traffic conditions even without an advanced

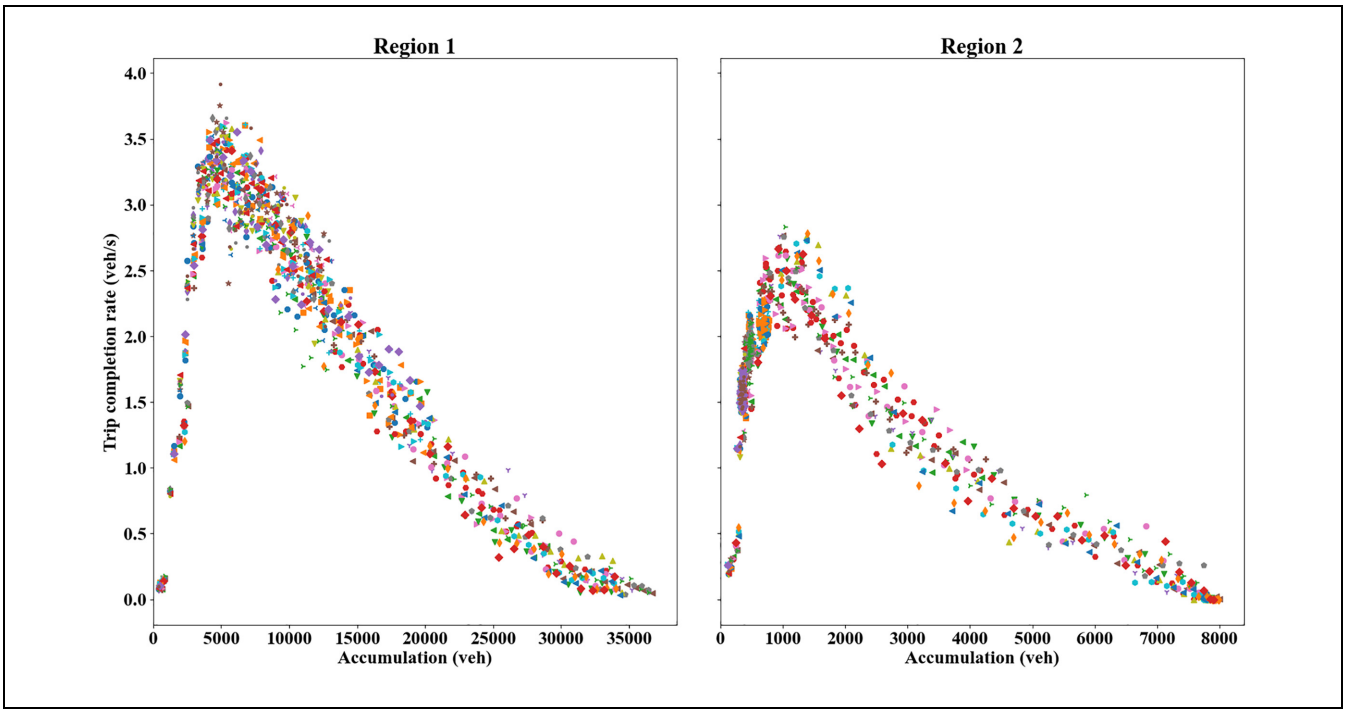

Figure 3. Regional trip completion rate-accumulation relationships (macroscopic fundamental diagram): Region 1 (*left*) and Region 2 (*right*).
Note: Each symbol represents a random seed in the simulation.

solution mechanism. As previously noted, the cutoff points used by I-GC are higher than the critical accumulations and set to $n_{1,c1} = 5000, n_{1,c2} = 8000, n_{2,c1} = 1000, \quad n_{2,c2} = 1600$. These values have been observed to help avoid excessively long queues at the perimeter control intersections while also suitable to mitigate the severe congestion in the city center. Similarly, observations on the simulations have informed proper green time settings, which are $g_{min} = 0s$, $g_{mid} = 3s, g_{max} = 27s$. The minimum green time g_{min} is adopted when a region is severely congested with accumulations in $[n_{i,c2}, n_{i,jam}]$. Since the cutoff point $n_{i,c2}$ is noticeably larger than the critical values, all vehicle entries into the region should be prevented to restore the regional operations and avoid further performance degradations. The medium green time g_{mid} is used when a region is operating in critical flow (with accumulation in $[n_{i,c1}, n_{i,c2}]$). Within this range, regional congestion has built up but the average flow is at a relatively high level. Therefore, it is reasonable to reduce the amount of inflow into the region. Once the metering decision is made, queues will form at the upstream locations of all perimeter control intersections. Observations on the simulation show that, when the queuing vehicles are allowed entry in a time step, they discharge at a very high rate (not necessarily the saturation flow, because of factors such as startup loss time as well as the mismatch between simulated vehicle dynamics and theoretical models) and 3 s green time at the perimeter control intersections would correspond to an aggregate inflow rate of 5,760 vehicles per hour. Thus, g_{mid} is set to 3 s as it helps avoid long queues and leads to reduced entry rate that is much smaller than the traffic demands (see Figure 2). It also provides a control buffer between complete metering and admitting of transfer flows that helps smooth the traffic patterns around the critical accumulations, and improves the applicability of vanilla greedy control in a microsimulation setting. Finally, the maximum green time used when the region is in free flow is set to 27 s, followed by a 3 s change interval. Note that the cutoff points and green time settings are mainly obtained via observing the simulations, rather than from mathematical derivations. As such, they may not be analytically optimal. However, deriving the best form of I-GC implementation is beyond the focus of this work and, as will be shown in the results, the I-GC policy adopting these values can substantially improve the operational performance of the overall network.

In this work, multiple random seeds are used in the simulation to enhance realism where each seed corresponds to a specific traffic pattern (e.g., vehicle routing, speed distribution, times of vehicle insertion). These seeds are used both by the benchmarking methods and the RL controller. Concretely, for the RL controller, a

seed will be randomly picked every time a simulation is run during the training process so that it can learn a robust policy against simulation randomness. In contrast, the benchmarking methods are directly applied to the simulation with all seeds, as they are not learning-based approaches and adopt fixed policies. As such, each random seed is associated with a constant control outcome for these methods, and their performances among multiple random seeds will be expressed as narrow horizontal bands.

Simulation Results: Effectiveness

This section presents the simulation results on evaluating the effectiveness of the RL method. Adopting the baseline demands in Figure 2, the simulation starts from an empty network and the first 90 s is used as a warmup period during which the network operations are not recorded. The presented RL method, as well as the benchmarking I-GC and NC policies, are applied, and the realized cumulative trip completion (CTC) is shown in Figure 4, where the solid line and shaded area separately indicate the mean and 95% confidence interval. Note, the benchmarking methods are not learning-based approaches and adopt fixed policies based on the environment information (e.g., regional accumulations for I-GC). Therefore, their "learning curves" appear as horizontal bands and do not exhibit an upward trend, while the shaded areas represent the variation over random seeds.

As shown in Figure 4, the I-GC and RL controller can achieve significantly higher CTCs than NC. While the design ideas are intuitive, the I-GC policy can effectively prevent oversaturation in the city center and realize a more productive congestion distribution within the network. The RL controller, with the ability to learn from interactions with the microsimulation, can also yield control policies that lead to increasing CTCs over its learning course. This learning ability is even more notable given that multiple random seeds are used which can affect the realized demand patterns and vehicle behaviors. As a side effect, though, the abundance of simulation randomness also leads to noticeable fluctuations in its learning curves.

To see how perimeter control improves the regional level traffic operations, the accumulations and average speeds are compared among the control methods. The accumulations are obtained by counting the number of vehicles in both regions, while the average speed is the average of lane-level speed weighted by their lengths. The profiles of (regional or network) accumulations and average speed are presented in Figures 5 and 6, respectively, where the cutoff points used by I-GC are also plotted in Figure 5*a*. As can be seen, under NC, the demands from the periphery would enter the city center

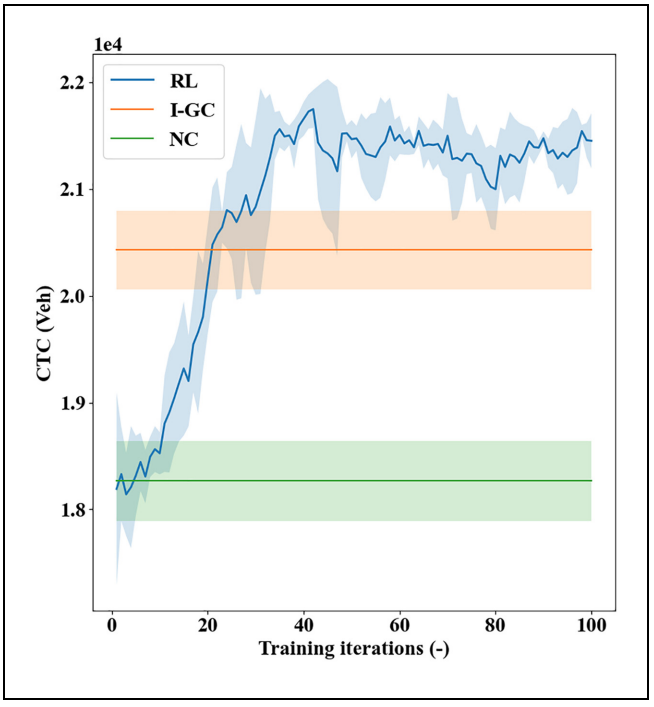

Figure 4. Cumulative trip completion (CTC) realized by different methods under the baseline demand.
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning; solid line = the mean; shaded area = 95% confidence interval.

almost unrestrictedly. As such, the Region 2 accumulation increases rapidly, and the congestion leads to a consistently decreasing average speed that plateaus around 3,000 s. Even in the recovery period (3,600–5,400 s), Region 2 cannot restore its operations from such congestion when there is no new traffic demand, which suggests the formation of traffic gridlock. In comparison, under the I-GC policy, Region 2 accumulation is well regulated between the two cutoff points adopted, whereas Region 1 is significantly more congested than under NC. This metering policy considerably improves the average speed of Region 2 at the cost of decreased average speed in Region 1. In addition, the strictness in metering also leads to fewer trips completed in the network (see Figures 4 and 5*b* where the peak value of network accumulation is smaller under RL than I-GC). On the other hand, the RL method achieves a middle ground between NC and I-GC in that it effectively mitigates congestion in the city center compared with NC while, in the meantime, not hindering vehicle transfer too much, to realize the highest trip completion. Furthermore, Figure 5 indicates that perimeter control (RL or I-GC) can produce a congestion distribution where both regions are trending toward clearance of vehicles at the end of the simulation, while the network (in particular Region 2) remains severely congested under NC.

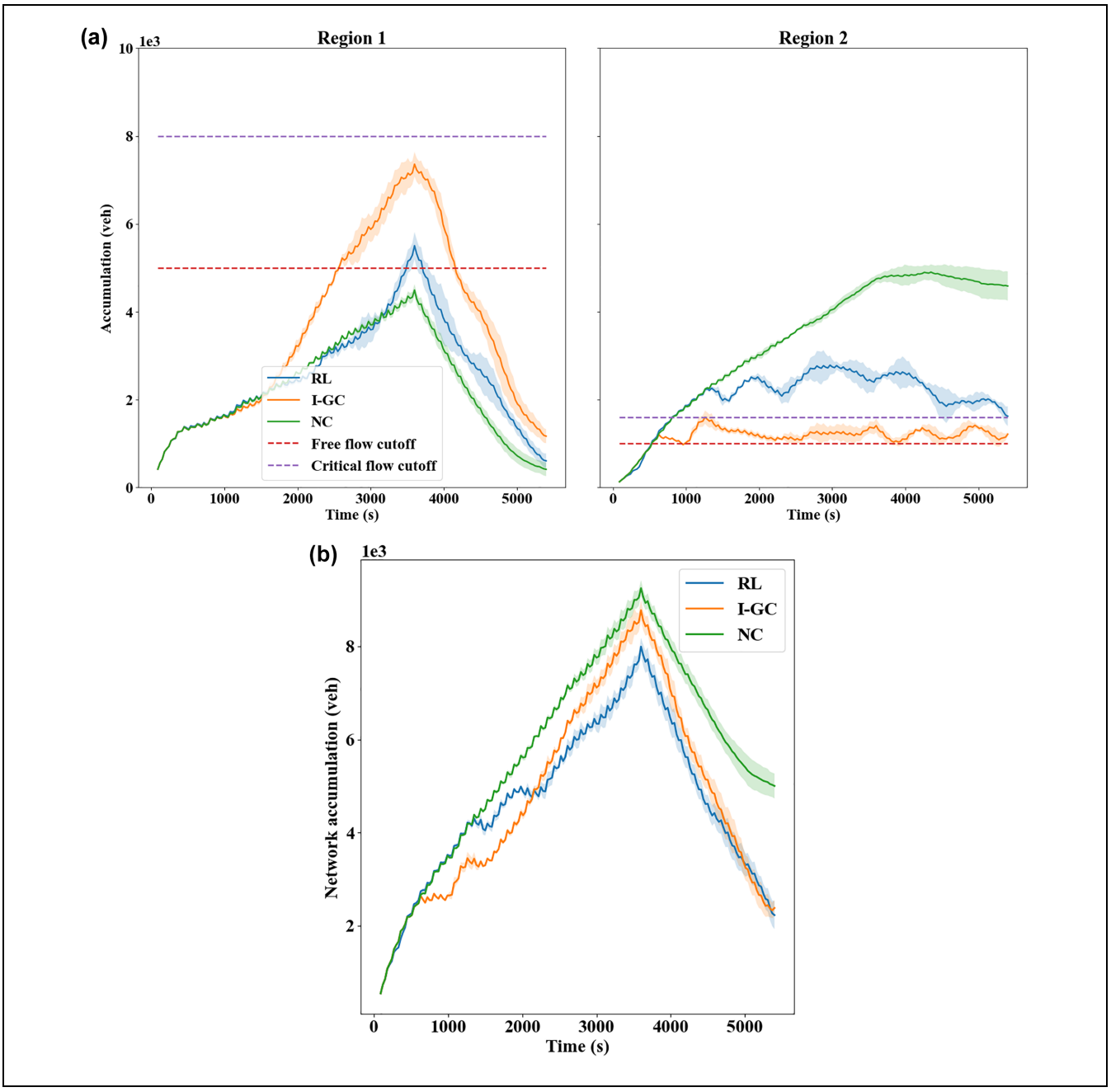

Figure 5. (*a*) Evolution of accumulations in Region 1 (*left*) and Region 2 (*right*) and (*b*) Evolution of accumulations in the network.
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning.

To further compare the capability of the methods on maximizing the network throughput, the trip completion plot is provided in Figure 7, where the total travel time (TTT) difference can be calculated as the areas between the curves. As can be seen, both RL and I-GC realize higher CTCs than FT, where the mean improvements are, respectively, 16.9% and 11.8%. However, despite the higher trip completion, the I-GC policy results in a large TTT than FT (by 1.835×10^6 s on average). In contrast, the RL method achieves the smallest TTT that is, on average, 1.949×10^6 s smaller than FT. Along with the highest trip completion, this showcases the superiority of the RL method at throughput maximization.

A few additional remarks are provided here. First, the city center is destination-loaded with large traffic demands from the periphery area. Therefore, perimeter control is the most helpful if it helps prevent Region 2 from becoming oversaturated. In this regard, both the I-GC and the RL controller can effectively realize this

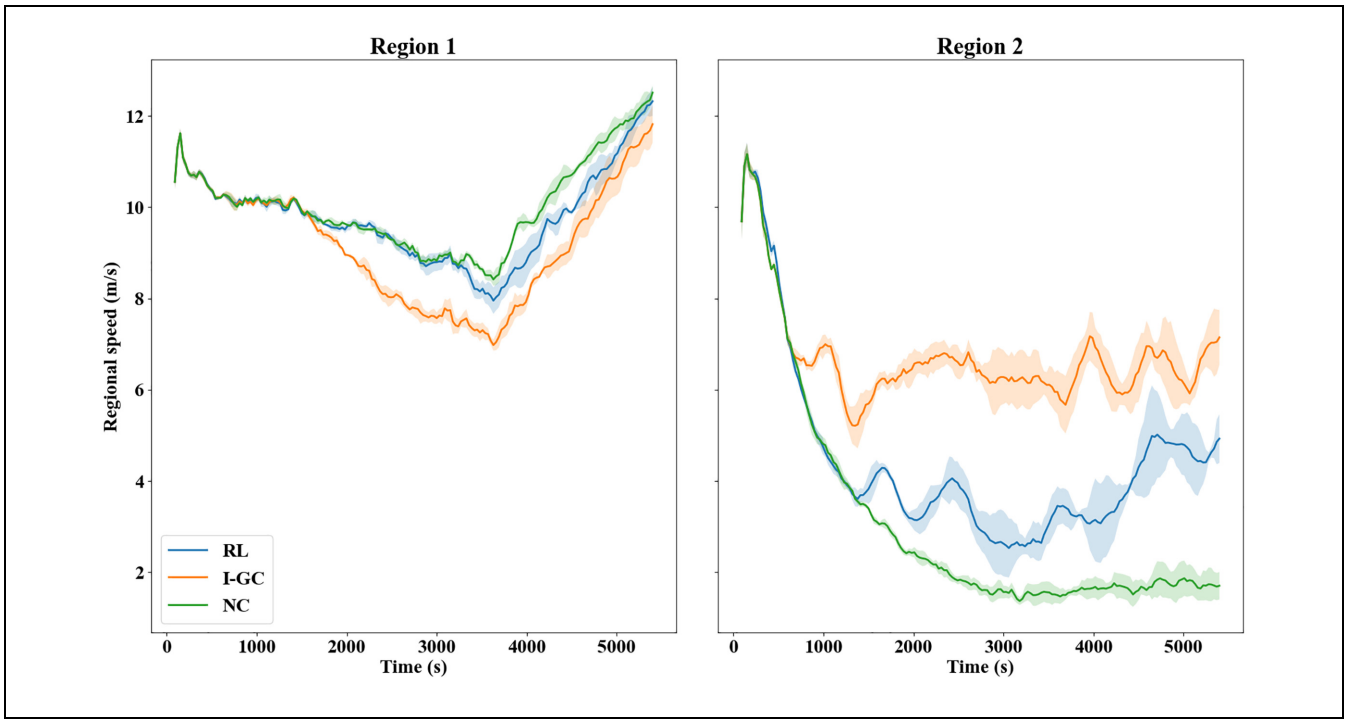

Figure 6. Average speed: Region 1 (*left*) and Region 2 (*right*).
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning.

control objective, in that Region 2 has substantially improved operational performance compared with NC. Second, the RL controller presented in this paper can effectively regulate traffic and consistently achieve higher CTC than the I-GC policy, despite differences in the control outcomes. Third, the RL controller adopts the same action designs as I-GC to fairly compare the two methods. As a result, the simulation outcomes could exhibit noticeable fluctuations over time (see Figure 6). Foreseeably, a more flexible action space design could potentially yield smoother network operations, as demonstrated in (*30*).

Real-life measurements of traffic states (e.g., vehicle accumulation) are often subject to noises because of factors such as sensor malfunction. Thus, to evaluate the learning robustness of the presented RL controller, measurement noise of regional accumulations is examined. Here, the form of measurement noise considered is (similar to (*27*, *33*):

$$\tilde{n}_i = n_i + \mathbb{N}\left(0, \delta^2\right) \tag{5}$$

where
n_i = the true accumulation value in the environment,
$\tilde{n}_i$ = the measured accumulation value received by the controllers, and
$\mathbb{N}\left(0, \delta^2\right)$ = a mean-zero normal distribution with standard deviation δ.

The control methods receive inaccurately measured accumulations, while the microsimulation environment maintains the accurate accumulation values. Therefore, a mismatch exists between the actual congestion level and the controller's perception, which would affect the control policy should the controllers act on accumulation information. In other words, the RL method and I-GC policy are subject to the measurement noise, while NC is not. As such, the realized control outcomes remain the same under NC.

In this work, two levels of measurement noise with standard deviation of 25 and 50 are examined: the realized CTC curves during the training processes are presented in Figure 8. As can be confirmed, the NC method has performances invariant to the measurement noise, as its policy does not build on any accumulation information. The performances of I-GC are affected, but it still significantly outperforms NC with increased CTC. More importantly, the presented RL method can learn to achieve higher CTCs than I-GC in both levels of measurement noise. As the level of noise increases, the fluctuations in the learning curves also increase. Further, the average CTC improvement of RL over I-GC is higher with noise 50 than under noise 25, which showcases the RL controller's robustness to learn against increasingly noisy measurements. However, note that the measurement noise cannot be overly high as it may disrupt the controller's ability to regulate traffic (e.g., the controller

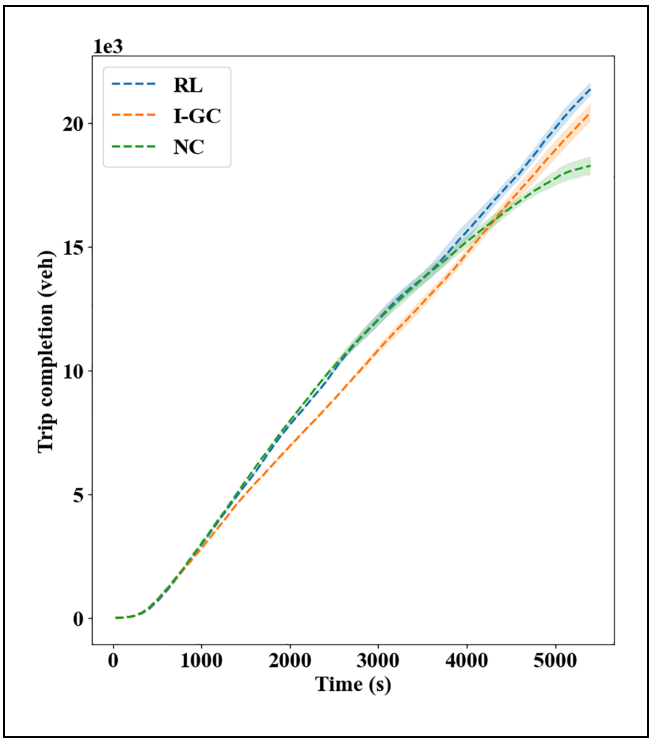

Figure 7. Trip completion plot.
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning.

might over-perceive the congestion in the environment heavily and meter the transfer flows needlessly, thus reducing trip completion). Overall, these results suggest that the presented RL controller is effective and robust, even with sizable measurement noise.

Simulation Results: Transferability

Transferability is examined here by applying the pre-trained RL controller with the baseline demand to different demand scenarios, driving behaviors, or both. Note that homogeneous transfer learning is considered to avoid negative transfer, so the network settings are not altered (as the dynamics underlying the problem would otherwise be significantly affected). Also, the premise of transfer learning is the source and target problems share adequate commonalities (as opposed to, say, models trained for image classification applied to natural language processing). It should thus be expected that the more different the two scenarios are, the less promising the transferred performances will be, though control benefits may still be achievable. Further, the controller has already internalized knowledge on how to conduct perimeter control based on the traffic conditions; therefore, during adaptation, less learning and exploration are needed. Finally, it is worth reiterating that these tests are

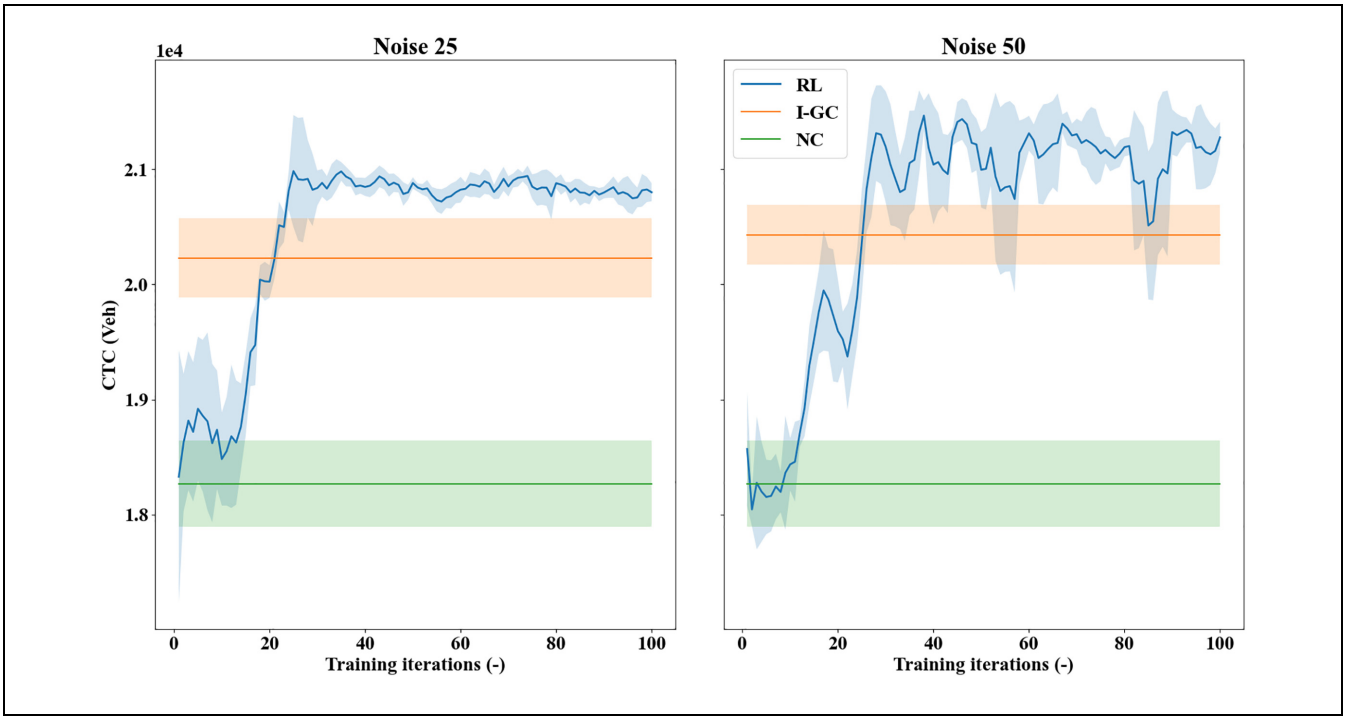

Figure 8. Realized cumulative trip completion (CTC) curves by different methods under noisy measurements: noise 25 (*left*) and noise 50 (*right*).
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning.

meant to showcase the ability of the RL controller to quickly adapt its policy to unseen traffic patterns by utilizing its knowledge learned from the source problem and continuing its learning with online data feed. This is important for real-life applications where training the controller from scratch may be expensive or even prohibitive.

The first scenario considers a non-uniform demand pattern from the periphery to the city center, while the other demands remain unchanged. This demand simulates a commute pattern where one side of the periphery area is more populous than the other. As an example, the west side of the periphery is assumed to have the most traffic demands into the city center (see Figure 9*a*). Further, this demand change is coupled with worse driving behaviors (e.g., less adaptive routing and higher driver imperfection), which was done to mimic real-world driving conditions where human drivers exhibit more variation in their driving performance than is typically modeled in simulation (*54*). For this traffic scenario, NC and I-GC are applied with numerous random seeds and, to better demonstrate transferability, an RL controller is also trained from scratch (with same parameters as used for the baseline demand) to compare with the transferred RL controller. The CTC achieved by different methods are provided in Figure 9*b*, where the transferred RL controller is denoted by "RL (Transfer)." The transferred RL controller is only trained for 50 iterations, as it is merely adapting its pretrained action value function. As can be seen, the RL controller, when trained from scratch, can still realize promising control benefits that are generally better than the I-GC policy. This indicates the applicability of the RL controller to different scenarios, more so considering the same parameters are used. Comparatively, the RL controller transferred with pretrained action value function learns at a faster rate from a substantially better starting point and achieves promising control performances within 20 iterations. With newly collected experiences, it can quickly adapt its action value functions to the different traffic scenario, even with less exploration of the environment and learning update. Importantly, these results indicate the RL controller, despite trained in an environment with better simulated driving behavior, can transfer its knowledge and quickly adapt its action-taking policy to a setting that is more representative of reality, which showcases its significant potential for practical application.

To evaluate the transferability of the RL controller to a setting with more dissimilar travel demands, a second traffic scenario is tested that considers an abrupt demand increase, as also examined in (*15*, *29*). In this scenario, the traffic demands from the periphery to the city center

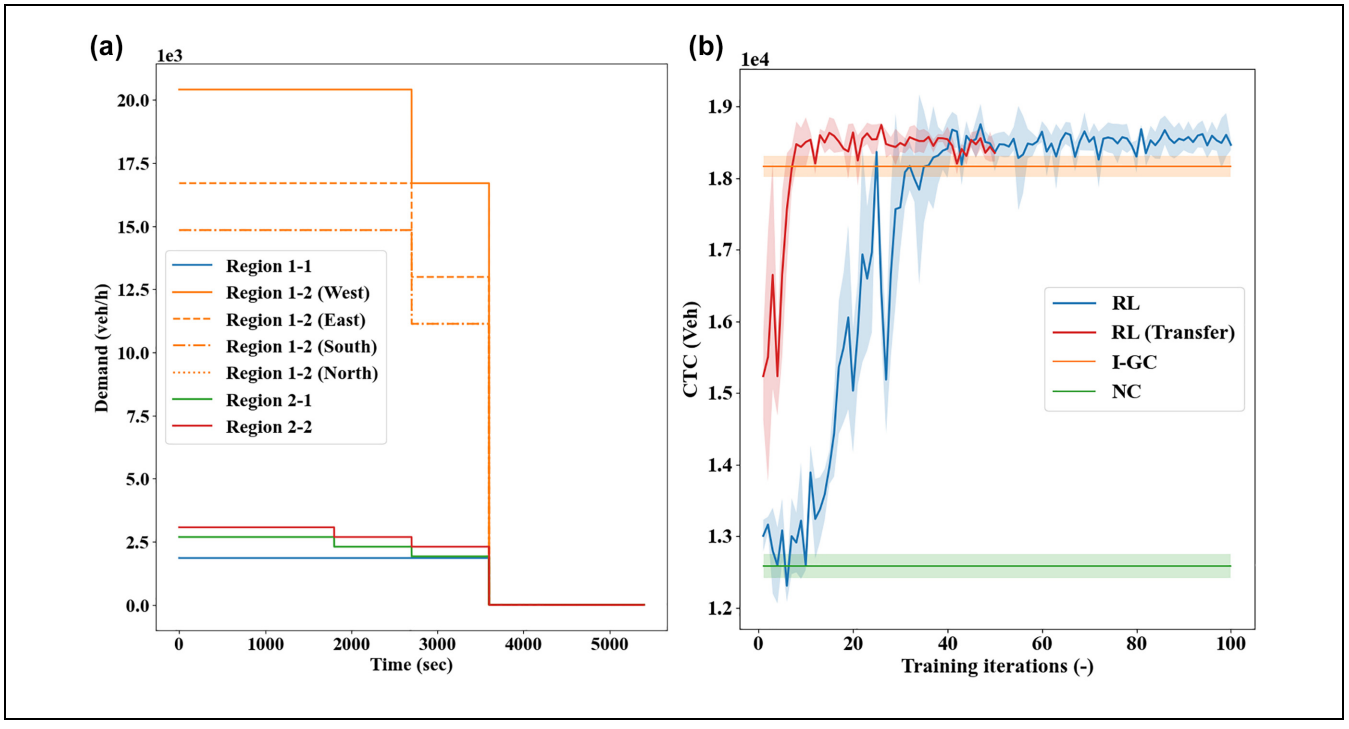

Figure 9. Transferability scenario 1: (*a*) demand profile and (*b*) realized cumulative trip completion (CTC) curves.
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning; RL (Transfer) = transferred RL controller.

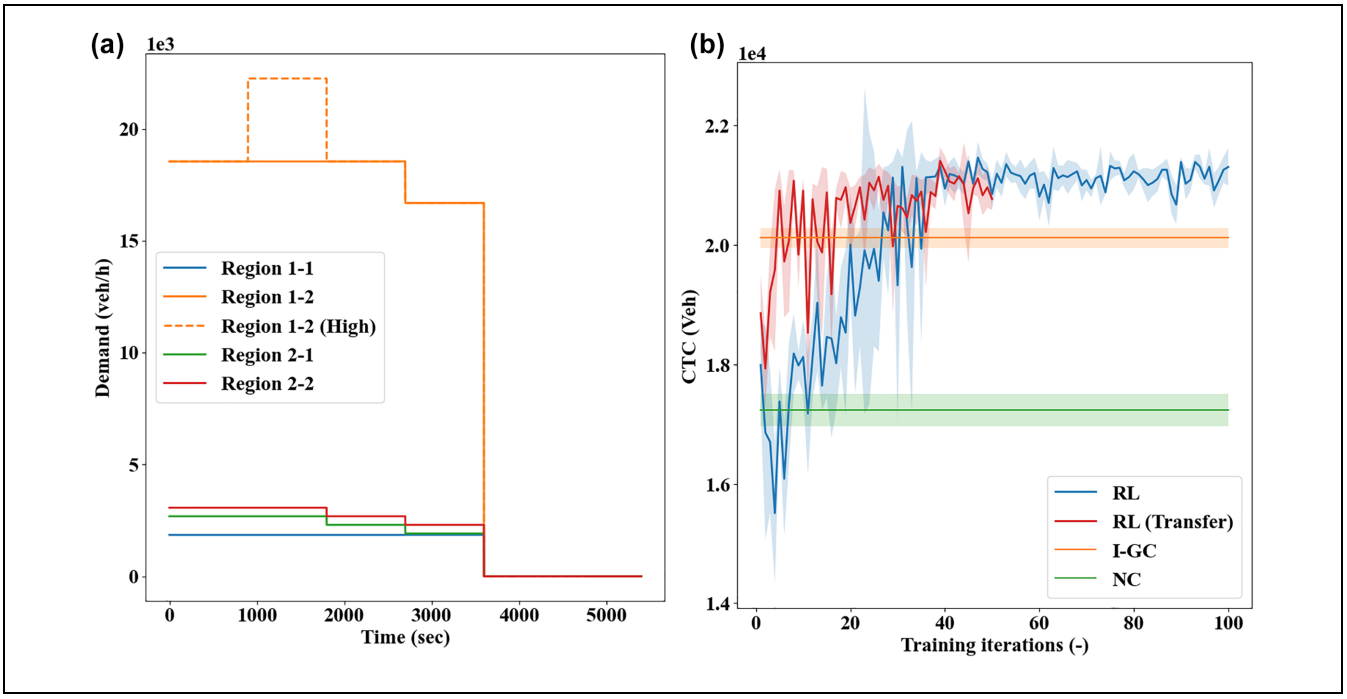

Figure 10. Transferability scenario 2: (*a*) demand profile and (*b*) realized cumulative trip completion (CTC) curves.
Note: I-GC = improved greedy controller; NC = no control; RL = reinforcement learning; RL (Transfer) = transferred RL controller.

have increased intensity from the baseline demand for a duration of 15 min (see Figure 10*a*). The RL controller pretrained with the baseline demand is applied to this scenario, in comparison with the NC, I-GC, and an RL controller trained from scratch. The CTC curves are presented in Figure 10*b*. As can be seen, the RL controller, when trained from scratch, can still learn control policies that are notably better than the I-GC, despite increased demand dissimilarity. The transferred RL controller with pretrained action value function can also achieve higher CTC than the I-GC policy, yet its learning performances are not as desirable as in the previous case (the CTC curves are noisier than in Figure 9). This, however, is expected, as increased difference between the pretrained and applied scenarios would require more learning and exploration, which may not be achievable within 50 iterations. Nevertheless, the CTC obtained by the transferred RL controller increases more rapidly than the one trained from scratch in the first 50 iterations, which again exemplifies its transferability.

Concluding Remarks

This paper studies the classical two-region perimeter control problem in a microsimulation environment. An RL-based perimeter controller is presented, with its effectiveness and transferability comprehensively demonstrated in a microsimulation environment. Importantly, the ability of a data-driven perimeter control method to transfer its learned knowledge from one setting to another and to quickly adapt its control policy with newly collected data is critical in real life applications, where training a controller from scratch may be costly or even impossible. Transferability of perimeter control methods has long been neglected, and the few recent works that considered transferring the learned policies have not examined the ability of the methods to keep learning with continued data feed. Further, no other studies that considered transferability have utilized the more realistic microsimulation environment but, instead, adopted simpler numerical simulations. This work thus bridges several research gaps and strengthens the existing literature on data-driven methods for perimeter control.

The simulation results provided in this work suggest the presented RL controller can consistently learn control policies that are superior to I-GC. More importantly, it can readily transfer its learned knowledge to simulation settings with different traffic demands, driving behaviors, or both, and keep learning from newly collected data to produce more promising control strategies. These results showcase the real-world application potential of data-driven methods for perimeter control. In particular, the RL controller can be trained offline in a microsimulation environment with an estimate of the real traffic demands, assuming proper driving behaviors; then, at the time of application, continued streams of traffic data can be collected and fed to the controller to adjust its policy. In this

manner, a smaller amount of data will be needed (as the controller has already internalized knowledge in the pre-train process) whereas the controller can adapt at a faster rate than if trained from scratch. Note that the application process can be carried out in real time as the action-taking only involves a forward pass using the parameters θ_t and less learning update is required to adapt the control policy.

Opportunities for future extensions exist. First, different forms of perimeter control implementation should be investigated, for example setting the green ratios or the allowable ratios of transfer flows (*17*, *19*, *29*, *30*). Also, it is worth examining if a more flexible action space could further improve the control benefits. Moreover, it is a research priority to evaluate RL-based perimeter control methods using a larger-scale urban network with microsimulation.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: D. Zhou, V. Gayah; data collection: D. Zhou, V. Gayah; analysis and interpretation of results: D. Zhou, V. Gayah; draft manuscript preparation: D. Zhou, V. Gayah. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by NSF Grant CMMI-1749200.

ORCID iD

Vikash V. Gayah https://orcid.org/0000-0002-0648-3360

References

1. Lowrie, P. R. *Scats: The Sydney Coordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic*. Roads and Traffic Authority NSW, Darlinghurst, NSW Australia, 1990.
2. Robertson, D. I., and R. D. Bretherton. Optimizing Networks of Traffic Signals in Real Time—The SCOOT Method. *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, 1991, pp. 11–15. https://doi.org/10.1109/25.69966
3. Godfrey, J. W. The Mechanism of a Road Network. *Traffic Engineering & Control*, Vol. 11, No. 7, 1969, pp. 323–327.
4. 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
5. 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.
6. Loder, A., L. Ambühl, M. Menendez, and K. W. Axhausen. Understanding Traffic Capacity of Urban Networks. *Scientific Reports*, Vol. 9, No. 1, 2019, pp. 1–10. https://doi.org/10.1038/s41598-019-51539-5
7. 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
8. 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
9. 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
10. 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
11. 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
12. 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
13. 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
14. 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
15. 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
16. 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

17. 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
18. 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
19. 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
20. 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
21. 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
22. Ampountolas, K., N. Zheng, and N. Geroliminis. Macroscopic Modelling and Robust Control of Bi-Modal Multi-Region Urban Road Networks. *Transportation Research Part B: Methodological*, Vol. 104, 2017, pp. 616–637. https://doi.org/10.1016/j.trb.2017.05.007
23. 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
24. 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
25. 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
26. 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*, Vol. 21, No. 7, 2019, pp. 1–12. https://doi.org/10.1109/tits.2019.2921381
27. 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
28. 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
29. Chen, C., Y. P. Huang, W. H. K. Lam, T. L. Pan, S. C. Hsu, A. Sumalee, and R. X. Zhong. Data Efficient Reinforcement Learning and Adaptive Optimal Perimeter Control of Network Traffic Dynamics. *Transportation Research Part C: Emerging Technologies*, Vol. 142, 2022, p. 103759. https://doi.org/10.1016/J.TRC.2022.103759
30. Su, Z. C., A. H. F. Chow, C. L. Fang, E. M. Liang, and R. X. Zhong. Hierarchical Control for Stochastic Network Traffic With Reinforcement Learning. *Transportation Research Part B: Methodological*, Vol. 167, 2023, pp. 196–216. https://doi.org/10.1016/J.TRB.2022.12.001
31. 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.
32. Zhou, D., and V. V. Gayah. Improving Deep Reinforcement Learning-Based Perimeter Metering Control Methods With Domain Control Knowledge. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2677, No. 7, 2023, pp. 384–405. https://doi.org/10.1177/03611981231152466
33. Zhou, D., and V. V. Gayah. Scalable Multi-Region Perimeter Metering Control for Urban Networks: A Multi-Agent Deep Reinforcement Learning Approach. *Transportation Research Part C: Emerging Technologies*, Vol. 148, 2023, p. 104033. https://doi.org/10.1016/J.TRC.2023.104033
34. Varaiya, P. Max Pressure Control of a Network of Signalized Intersections. *Transportation Research Part C: Emerging Technologies*, Vol. 36, 2013, pp. 177–195. https://doi.org/10.1016/j.trc.2013.08.014
35. Keyvan-Ekbatani, M., X. Gao, V. V. Gayah, and V. L. Knoop. Traffic-Responsive Signals Combined With Perimeter Control: Investigating the Benefits. *Transportmetrica B: Transport Dynamics*, Vol. 7, No. 1, 2019, pp. 1402–1425. https://doi.org/10.1080/21680566.2019.1630688
36. Tsitsokas, D., A. Kouvelas, and N. Geroliminis. Two-Layer Adaptive Signal Control Framework for Large-Scale Dynamically-Congested Networks: Combining Efficient Max Pressure with Perimeter Control. *Transportation Research Part C: Emerging Technologies*, Vol. 152, 2023, p. 104128. https://doi.org/10.1016/J.TRC.2023.104128
37. Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, and E. Wiebner. Microscopic Traffic Simulation Using SUMO. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Vol. 2018, 2018, pp. 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938
38. Girault, J. T., V. V. Gayah, S. I. Guler, and M. Menendez. Exploratory Analysis of Signal Coordination Impacts on Macroscopic Fundamental Diagram. *Transportation Research Record: Journal of the Transportation Research Board*, 2016. 2560: 36–46. https://doi.org/10.3141/2560-05
39. Cascetta, E., A. Nuzzolo, F. Russo, and A. Vitetta. *A Modified Logit Route Choice Model Overcoming Path Overlapping Problems. Specification and Some Calibration Results for Interurban Networks*. In *Transportation and Traffic*

Theory. *Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France*, 1996.
40. 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
41. Ding, H., Y. Zhang, X. Zheng, H. Yuan, and W. Zhang. Hybrid Perimeter Control for Two-Region Urban Cities With Different States. *IEEE Transactions on Control Systems Technology*, Vol. 26, No. 6, 2018, pp. 2049–2062. https://doi.org/10.1109/TCST.2017.2746061
42. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, et al. Human-Level Control through Deep Reinforcement Learning. *Nature*, Vol. 518, No. 7540, 2015, pp. 529–533. https://doi.org/10.1038/nature14236
43. van Hasselt, H., A. Guez, and D. Silver. Deep Reinforcement Learning With Double Q-Learning. *Proc., AAAI Conference on Artificial Intelligence*, Vol. 30, No. 1, 2015. https://doi.org/10.1609/aaai.v30i1.10295.
44. Horgan, D., J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver. *Distributed Prioritized Experience Replay*. arXiv preprint arXiv:1803.00933, 2018.
45. Sutton, R. S., and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.
46. van Hasselt, H. *Double Q-Learning*. Advance in Neural Information Processing Systems, 2010.
47. 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
48. Kingma, D. P., and J. L. Ba. *Adam: A Method for Stochastic Optimization*. arXiv preprint arXiv:1412.6980, 2015.
49. Tirinzoni, A., R. Rodriguez Sanchez, and M. Restelli. Transfer of Value Functions via Variational Methods. *Advances in Neural Information Processing Systems*, Vol. 31, 2018.
50. Liu, Y., Y. Hu, Y. Gao, Y. Chen, and C. Fan. Value Function Transfer for Deep Multi-Agent Reinforcement Learning Based on N-Step Returns. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 457–463.
51. Da Silva, F. L., and A. H. Reali Costa. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *Journal of Artificial Intelligence Research*, Vol. 64, 2019, pp. 645–703. https://doi.org/10.1613/jair.1.11396
52. Zhu, Z., K. Lin, A. K. Jain, and J. Zhou. *Transfer Learning in Deep Reinforcement Learning: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
53. Weiss, K., T. M. Khoshgoftaar, and D. D. Wang. A Survey of Transfer Learning. *Journal of Big Data*, Vol. 3, No. 1, 2016, pp. 1–40. https://doi.org/10.1186/S40537-016-0043-6/TABLES/6
54. Krauss, S. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. 1998.