Transportation Research Part C 116 (2020) 102661

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework

Jiawei Wang, Lijun Sun*

*Department of Civil Engineering, McGill University, Montreal, QC H3A 0C3, Canada*

ARTICLE INFO

*Keywords:*
Bus bunching
Dynamic holding
Multi-agent system
Deep reinforcement learning
Policy proximal optimization

ABSTRACT

Bus bunching has been a long-standing problem that undermines the efficiency and reliability of public transport services. The most popular countermeasure in practice is to introduce static and dynamic holding control. However, most previous holding control strategies mainly consider local information with a pre-specified headway/schedule, while the global coordination of the whole bus fleet and its long-term effect are often overlooked. To efficiently incorporate global coordination and long-term operation in bus holding, in this paper we propose a multi-agent deep reinforcement learning (MDRL) framework to develop dynamic and flexible holding control strategies for a bus route. Specifically, we model each bus as an agent that interacts with not only its leader/follower but also all other vehicles in the fleet. To better explore potential strategies, we develop an effective headway-based reward function in the proposed framework. In the learning framework, we model fleet coordination by using a basic actor-critic scheme along with a joint action tracker to better characterize the complex interactions among agents in policy learning, and we apply proximal policy optimization to improve learning performance. We conduct extensive numerical experiments to evaluate the proposed MDRL framework against multiple baseline models that only rely on local information. Our results demonstrate the superiority of the proposed framework and show the promise of applying MDRL in the coordinative control of public transport vehicle fleets in real-world operations.

## 1. Introduction

In public transport systems, bus bunching refers to the phenomenon where a group of two or more buses arrives at the same bus stop at the same time. Bus bunching has been a long-standing operational problem in urban public transport systems, and it is a primary issue that concerns transit users and affects our perception of service reliability and service efficiency. As first reported by Newell and Potts (1964), bus services are born unstable and they are in nature susceptible to bus bunching due to the inherent uncertainties in service operation. On the one hand, the uncertainty in the number of boarding/alighting passengers increases the variability of bus dwell time. On the other hand, the time-varying urban traffic conditions (e.g., congestion and signal control) and the differences in bus driving behavior also introduce considerable uncertainty in travel time. In real-world operations, a tiny disruption could result in multiple buses falling behind schedule. When this happens, the delayed buses in general will encounter more waiting passengers accumulated during the waiting period, which in turn will result in longer dwell times and further delays.

Bus bunching has several adverse effects on public transport operation. As bus bunching coming into being, first, it results in increased headway variability, and thus for transit users, both waiting time and travel time become longer; Second, as the system

* Corresponding author at: 492-817 Sherbrooke Street West, Macdonald Engineering Building, Montreal, Quebec H3A 0C3, Canada.
*E-mail addresses:* jiawei.wang4@mail.mcgill.ca (J. Wang), lijun.sun@mcgill.ca (L. Sun).

https://doi.org/10.1016/j.trc.2020.102661
Received 29 November 2019; Received in revised form 16 March 2020; Accepted 11 April 2020

operates, the buses tend to have unbalanced occupancy rates because most passengers prefer to board the first bus arrived, in particular when passengers do not have any information about the arrival of the next bus. Therefore, bus bunching eventually causes ineffective use of the supply of public transport services. In the long term, bus bunching will also affect users' perception of service reliability and in turn discourage the use of public transport as a key solution to sustainable transportation.

Given the considerable negative impact of bus bunching, extensive efforts have been made to address this challenge and develop new control strategies, such as introducing stop-skipping as a real-time control solution to allow vehicles to catch up with schedule by skipping certain stops (Fu et al., 2003; Sun and Hickman, 2005), allowing bus drivers to adjust cruising speed of vehicles to maintain a regular headway (Daganzo and Pilachowski, 2011), introducing boarding restrictions/limits to control bus dwell time (Delgado et al., 2009, 2012; Zhao et al., 2016), implementing traffic signal priority (Estrada et al., 2016), and deploying substitution buses (Petit et al., 2018). However, these strategies are not widely adopted in practice: stop-skipping and boarding restriction will penalize waiting passengers and they will make passengers even more unsatisfied, and other strategies essentially require additional resources/infrastructure from the agencies/operators.

Despite the rapid development of new solutions, vehicle holding control (both static and dynamic) remains the most practical and most adopted strategy in real-world application (Cats et al., 2011; Wu et al., 2017b). As pointed out by Eberlein et al. (2001), bus holding control strategies not only make passengers less frustrated but also provide better maneuverability to fleet operation. Related studies on bus holding control can be mainly categorized into two classes: static headway/schedule-based control and dynamic control without a pre-specified headway or schedule. For static control, Zhao et al. (2006) developed a holding strategy that minimizes the passengers' expected waiting time under schedule-based control. Daganzo (2009) introduced a dynamic model that determines bus holding times at a route's control points by achieving target headway. However, static control has an inherent technical weakness, since the ideally achievable headway is neither static nor known in advance (Bartholdi and Eisenstein, 2012). To address these issues, dynamic bus holding methods have been developed recently. For example, Xuan et al. (2011) introduced a dynamic holding strategy based on virtual schedule, which turns out to be more efficient than conventional schedule-based methods. Bartholdi and Eisenstein (2012) proposed a dynamic bus holding strategy with a single control point, in which the headways between every two consecutive buses are dynamically self-equalizing. Although these strategies have shown satisfying applicability, there is still a large room to develop a better control policy. On the one hand, most aforementioned studies either focus on dealing with local control problems, or presented formulate as a mathematical problem on the centralized scheme requiring potentially heavy computations, which may limit the number of buses taken into account, thus it remains a challenging task to achieve near-optimal control for the fleet as a whole through these local control strategies (Chen et al., 2016). On the other hand, the traditional polices mainly overlook long-term effect of each single control and may develop short-sighted policies, it would be better to consider the subsequent impact from each control, therefore keep system as efficient as possible in the long run.

The recent progress in machine learning, in particular deep reinforcement learning (DRL), has shed new light on solving complex sequential decision-making problems in many domains. By integrating deep neural network with reinforcement learning, DRL not only has great generalization ability but also characterizes control feedback in long-term horizons (Mnih et al., 2015; Mnih et al., 2016; Schulman et al., 2017). To this end, urban traffic control (e.g., traffic signal and autonomous fleet) has recently become a perfect application area of DRL (Kuyer et al., 2008), and recent research has shown increasing interest to solve traditional traffic problems using DRL (see e.g., Li et al., 2016; Wu et al., 2017a; Aslani et al., 2017; Alesiani and Gkiotsalitis, 2018). Moreover, as most traffic control problems involve decision coordination among a group of decentralized entities (e.g., regional traffic signal control and fleet operation), multi-agent reinforcement learning (MRL) models have been developed to address the underlying computation challenges in real-world transportation systems (El-Tantawy et al., 2013). On this track, a particular application is to develop coordinative fleet control algorithms to address the aforementioned bus bunching problem. For example, Chen et al. (2016) recently presented a multi-agent Q-learning architecture for coordinated bus operation on a transit corridor. However, there are still some major drawbacks. First, the goal in their study is defined to match pre-specified headway, such a setting is not flexible enough to respond to the variations in transits system. Second, they have not taken full advantage of DRL, consequently, the method only explores policy within a limited state-action space. Besides, the learning procedure tends to be more computational expensive. Alesiani and Gkiotsalitis (2018) developed deep reinforcement learning to implement holistic holding control, but they also try to specify a target headway and have not taken into account the holding action of each agent explicitly. To address these issues, in this paper we propose a new multi-agent deep reinforcement learning (MDRL) framework to integrate global information and to develop efficient and reliable holding control policy considering the whole transit system. Our main contributions are:

- To exploit the advantage of DRL, we establish a MDRL framework to implement dynamic bus holding control with properly designed reinforcement learning elements, including local and global state definition, fine-grained action, and a dedicated reward function.
- To facilitate coordination among agents within the framework, we introduce a joint action tracker to approximate actions of each agent, as a supplement to the basic actor-critic scheme.
- To train each agent within the framework efficiently, we design a specific learning procedure based on proximal policy optimization (PPO), and enforce the learned policy on each bus in the transit system.
- To demonstrate superiority of the proposed framework, we develop a real-world transit simulator, on which quantitative analyses and baseline comparisons are conducted.

The remainder of the paper is organized as follows. In Section 2, we describe the bus bunching problem and holding control strategies in detail, and also introduce basic model assumptions. Section 3 describes the proposed MDRL framework and the

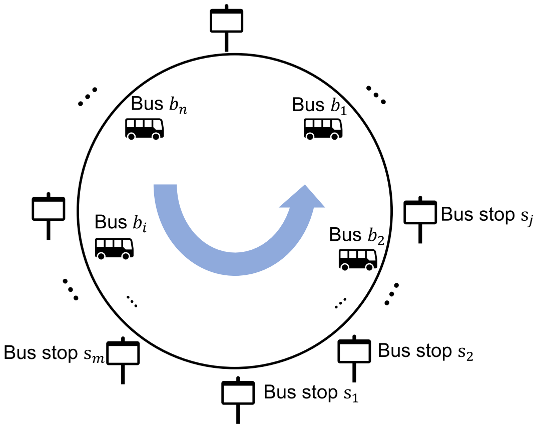


**Fig. 1.** Representation of bus corridor.

corresponding learning algorithm. We present our experiments and analyses in Section 4 and Section 5 concludes our study and discuses potential future study directions.

## 2. Problem statement

We follow the same modeling framework as in Daganzo (2009) and work with a stylized model of one-way bus corridor with $m$ uniformly distributed bus stops. The bus fleet of this corridor consists of $n$ buses with constant average speed $v$. We use $i$ and $j$ to index buses and bus stops, respectively. Fig. 1 shows the general setup of our framework. We refer to dynamic holding control as the policy that holds the bus at a control point (i.e., a bus stop) for certain time $\triangle T$. Below we introduce the assumptions and basic notations in this paper.

(a) We denote bus stops along the corridor by $s_j$ for $j = 1, 2, ..., m$. All bus stops are uniformly distributed on the corridor.
(b) We denote the $i$-th bus on the corridor by $b_i$ (for $i = 1, 2, ..., n$). All buses follow a sequential order, where vehicle $b_{i+1}$ follows vehicle $b_i$ for $i = 1, 2, ..., n - 1$ and vehicle $b_1$ follows vehicle $b_n$. Without loss of generality, we refer $b_{i+1}$ as the follower of $b_i$. The forward headway of bus $b_i$ to its leading vehicle is denoted by $h_i^-$ and the backward headway of bus $b_i$ to its follower is denoted by $h_i^+$. It should be noted that the backward headway of $b_i$ is the same as the forward headway of $b_{i+1}$: $h_i^+ = h_{i+1}^-$.
(c) When initializing the system, all buses are uniformly distributed with equal spacing. We impose a no-overtaking constraint and all vehicles cannot skip bus stops.
(d) When bus bunching happens at bus stop $j$, waiting passengers will board all arrived buses (i.e., the bunching cluster) with equal probability.
(e) The boarding and alighting processes start simultaneously and follow linear models of the number of passengers. The boarding time and alighting time per passenger are $\tau_b$ and $\tau_a$, respectively, for all bus stops. Buses cannot leave the stop until both the boarding and the alighting processes finish, and dwell time is determined by the maximum of the two values. We assume that bus capacity is unlimited.
(f) At each bus stop $s_j$, passenger arrival follows a Poisson process with rate $\lambda_j$. For a passenger waiting at bus stop $s_j$, her destination is randomly chosen from the following $\lfloor \frac{m}{2} \rfloor$ stops with uniform probability. Here, $\lfloor \cdot \rfloor$ represents the floor function, e.g., $\lfloor \frac{7}{2} \rfloor = \lfloor \frac{6}{2} \rfloor = 3$.
(g) Holding decisions are made after the whole dwell process (all boarding/alighting activities) finishes.

The assumptions above characterize the operation of the bus system and the interactions among all entities including buses, waiting passengers, and bus stops.

## 3. Methodology

### 3.1. Overall framework

Based on the aforementioned assumptions, we next describe the multi-agent deep reinforcement learning framework (see Fig. 2).

In this framework, each bus is running along sections of the corridor. The decisions for holding activities are made when a bus arrives at a bus stop, and the state observations are from the real-time information of the whole bus system. The reward feedback is calculated upon the bus arriving at the next stop, and then new holding decision process will be activated. Formally, we consider this dynamic process a multi-agent extension of Markov decision processes (MDPs) and refer to this process as Markov games (Littman, 1994). Essentially, a Markov game can be defined by a tuple:

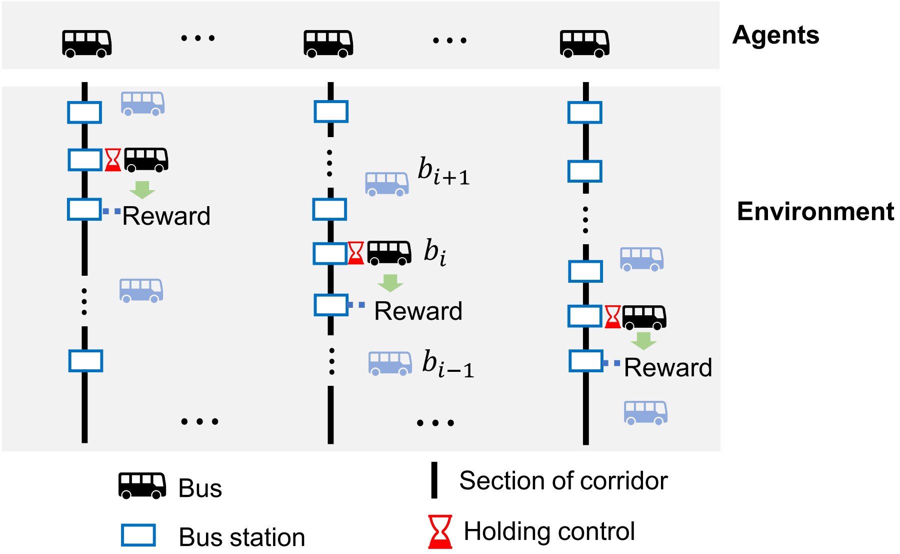


**Fig. 2.** Illustration of the multi-agent deep reinforcement learning framework for dynamic holding.

$$G = (N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma), \tag{1}$$

where $N$ denotes the number of agents, $\gamma$ denotes the discount factor, $\mathcal{S}$ and $\mathcal{A} = \{A_1,...,A_N\}$ denote the state space and the joint action space, respectively, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ represents the state transition function. We denote the set of reward functions associated with each agent $\mathcal{R} = \{R_1,...,R_N\}$, with each $R_i$ determined by the current state and the actions from all agents: $\mathcal{S} \times A_1, ..., \times A_N \mapsto R_i$. Finally, the policy to choose an action is given by the conditional probability $\pi_{\theta_i} = p(A_i|O_i)$, where $O_i$ is the observation of agent $i$ on current state $\mathcal{S}$.

Based on the aforementioned formulation, we consider all the $n$ ($n = N$) buses in the studied system as homogeneous agents. The components for reinforcement learning in this system are given below:

**State:** The state of the bus system should reflect the information on both buses and passengers. Assuming at time $t$ bus $b_i$ arrives at stop $s_j$, we categorize the current state observations into two types:

**State observation for agents:** We denote the state observation for bus $b_i$ by $X_i^t = [h_i^{t-}, h_i^{t+}, z_t/Z]$, where $h_i^{t-}$ and $h_i^{t+}$ are the forward headway and the backward headway at $t$, respectively, $z_t$ represents the number of waiting passengers at bus stop $s_j$ when bus $b_i$ arrives, and $Z$ is a normalizing constant.

**State observation for tracker:** We let $O^t = [X_1^t,...,X_n^t]$ denote the state observations. Tracker will approximate the actions of each agent in the system based on $O^t$, and thus keep track of the outcomes of actions from each agent without knowing their policies.

**Action:** We denote by $a_i^t$ the action of bus $b_i$ when arriving at bus stop $s_j$. The holding time/duration is formulated as

$$\triangle d_i^t = a_i^t \triangle T, \tag{2}$$

where $\triangle T$ is the maximum holding time and $a_i^t \in [0, 1]$ is a strength parameter. Here, $\triangle T$ is introduced to restrict the maximum holding duration and avoid too much intervention. Thus, $a_i^t = 0$ suggests no holding control is implemented. In particular, we consider a continuous $a_i^t$ in favor of exploring the near-optimal policy in an adequate action space. In multi-agent case, the join-action at time step $t$ will result in the state transitions $[X_1^{t+1}, ...X_N^{t+1}]$ at $t + 1$ step. That is to say, the joint action of each agent $[a_1^t, ..., a_N^t]$ will specify how long each bus holds at the arriving stop, which will have influence on the headway among buses and number of waiting passengers at each stop.

**Reward function:** As suggested by Bartholdi and Eisenstein (2012), a flexible and effective solution to avoid bus bunching is to reduce both the mean and the variance of headway at the same time. Inspired by this work, we follow the same objective from a distributed perspective. Specifically, we define the reward function associated with each bus $b_i$ serving stop $s_j$ at time step $t$ as

$$r_i^t = \exp(-|h_i^{t-} - h_i^{t+}|) + w\exp(-a_i^t), \tag{3}$$

where $w$ is a weight hyperparameter to be defined.

The reward function in Eq. (3) consists of two components. The first term $\exp(-|h_i^{t-} - h_i^{t+}|)$ clearly encourages headway equalization and the maximum value is obtained when $h_i^{t-} = h_i^{t+}$. The second term penalizes holding control decisions. This is due to the fact that any holding decisions will increase average headway in the system, and as a result on-board passengers also suffer from additional travel time. In other words, this term prevents the learning algorithm from making excessive control decisions. Overall, the goal of the reward function in Eq. (3) is to achieve vehicle coordination/synchronization with as few interventions as possible. In terms of computation, the reward function can also speedup policy exploration since agents will exploit the decentralized structure of the task (Kuyer et al., 2008).

When each agent $i$ arrives at stop $j$ at time step $t$, it observes state $X_i^t$ and determines holding action $a_i^t$, which will have impact on the state of the whole system. By the time this agent arrives at the next stop at time step $t + \Delta t$, it will receive next state observation $X_i^{t+\Delta t}$ and the reward $r_i^{t+\Delta t}$ which evaluates its last action. All the records $\{X_i^t, a_i^t, r_i^{t+\Delta t}, X_i^{t+\Delta t}\}$ collected during each entire simulation form a trajectory $Tr$ for that agent. The discounted future reward from each step $k$ is defined as $\sum_{t=k}^{|Tr|} \gamma^{t-k} r_t^i$, where $\gamma$ is discount factor. In this way we take into account future effect of each control action in the optimization problem. The ultimate goal is to find a policy for each agent that can maximize discounted future reward.

We next introduce how to formulate and solve the multi-agent deep reinforcement learning problem with proximal policy optimization (PPO).

### 3.2. Proximal policy optimization (PPO)

Considering that reinforcement learning is applied in finding an optimal policy in solving sequential decision-making problems, a natural objective is to maximize the discounted cumulative reward $\sum_t \gamma^t r_t$ in the long term (Sutton et al., 1998). A common solution is to perform policy gradient, which transforms the original problem into an optimization problem based on parameterized policy $\pi_\theta$ (Sutton et al., 2000):

$$J(\theta) = E_{a \sim \pi_\theta} [q_\pi(o_t, a_t) \pi_\theta(a_t | o_t)], \tag{4}$$

where state-action value $q_\pi(o_t, a_t)$ represents the expected cumulative reward if we select action $a$ given observation $o$ at time step $t$.

However, performing optimization on the stochastic policy is often unstable and it cannot guarantee monotonic policy improvement. To increase learning efficiency and make the train procedure more stable, Schulman et al. (2017) proposed proximal policy optimization (PPO), which uses a surrogate objective function to replace $J(\theta)$:

$$J_{\text{PPO}}(\theta) = E_{a \sim \pi_{\theta_{\text{old}}}} [\min(\delta_t(\theta) \widehat{A}_t, \text{clip}(\delta_t(\theta), 1 - \epsilon, 1 + \epsilon) \widehat{A}_t)], \tag{5}$$

where $\pi_\theta$ is the policy to train, and its duplication $\pi_{\theta_{\text{old}}}$ stores parameters before each update, $A_t$ is an estimator of the advantage function at time step $t$, $\delta_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$ is derived from importance sampling, $\epsilon$ is a hyper-parameter, clip stands for clipping based on the setting scale.

The basic intuition is that, if a policy currently shows poor performance (i.e., $\widehat{A}_t < 0$), it will be updated without imposing any constraints; otherwise, if $\widehat{A}_t > 0$, the adjustment on the policy should be limited. With this idea, the training procedure will be more robust and efficient.

#### 3.2.1. Multi-agent PPO with joint action tracker

Based on the multi-agent deep reinforcement learning scheme presented by Lowe et al. (2017), in this paper we propose multi-agent PPO with a joint action tracker (MPPO-T) to address bus bunching. The major contributions include:

- We suggest a decentralized actor neural network with local information for decision making (i.e., the parametrized policy). The decentralized critic neural network not only acts as state value estimator for each agent using local information but also tracks the actions from the joint action tracker.
- We introduce a deep neural network—the joint action tracker—to approximate action from each agent through supervised learning. In contrast to Lowe et al. (2017) that maintain multiple action estimators for each agent, we reduce the complexity by using a joint action tracker based on global information. This kind of scheme is critical, since we can avoid training agents with unknown interaction dynamics among agents' policies (Lowe et al., 2017).
- We apply PPO to update the actor neural network for a stable policy training procedure, and then a training algorithm is devised for MPPO-T to learn dynamic bus holding control.

As shown in Fig. 3, each agent maintains an actor (i.e., decision making) and a critic (i.e., policy evaluation), respectively. There exists only one global joint action tracker for the whole system. Essentially, the mechanism for each agent is based on the traditional actor-critic framework (Sutton et al., 1998). More specifically, for agent $i$, the decision making (i.e., $\pi(a_i^t | X_i^t; \theta)$) is based on the local observation $X_i^t$ from environment (i.e., transit system), particularly, the agent will take account of its forward headway and backward headway as well as the number of waiting passengers, then determine holding control period, and receive reward signal $r_i^t$ at its next stop. Besides, the critic is responsible for estimating state value (i.e., $\widehat{V}_i^t(X_i^t, \widehat{a}_1, \ldots, a_{i-1}^t, a_{i+1}^t, \ldots, \widehat{a}_N^t; \alpha_i)$) given local information $C_i^t$ plus joint action $\widehat{\boldsymbol{a}}(O^t; \phi)$ except its own action. Then, this evaluation is used by the actor to update its policy. Note that joint action is approximated by joint action tracker and it considers global information $O^t$ of transit system. In this way the critic can consider extra information about the approximated policies of other agents. As a result, the state evaluation of each agent's policy is based on the local environment, as well as other agent's decision. Therefore the coordination among the system is promised. In addition, the reason why the actor is limited to local information is that it is beneficial to avoid extra approximation during real-world application.

Formally, suppose that each agent $b_i$ experiences a length-$l$ trajectory $(X_i^1, a_i^1, r_i^1, X_i^2, ..., X_i^l, a_i^l, r_i^l)$, the corresponding optimization problem are as follows:

For actors, the goal is to learn the policy to enable the agent to obtain cumulative reward as large as possible. Utilizing the PPO, the objective function for agent $i$ is to be maximized subject to:

$$L_{\text{actor}}(\theta_i) = \sum_{t=0}^{l} \min(\delta_i^t(\theta_i)\hat{A}_i^t, \text{clip}(\delta_i^t(\theta_i), 1-\epsilon, 1+\epsilon)\hat{A}_i^t) \tag{6}$$

where $\gamma$ denotes the discount factor, $\hat{A}_i^t = \sum_{k=t}^{l} \gamma^t r_i^k - \hat{V}_i^k$ denotes advantage, $\delta_i^k = \frac{\pi(a_i^k \mid X_i^k;\theta_i)}{\pi(a_i^k \mid X_i^k;\theta_i^{\text{old}})}$ denotes the ratio between new policy and the old one, and $\theta_i$ represents parameters of critic for agent $i$.

For critics, the goal is to give an accurate evaluation on current policy (i.e., the output of the critic is the estimated state value for the agent), thus it can treated as supervised learning, where the objective function is to minimize the subject as follows:

$$L_{\text{critic}}(\alpha_i) = \sum_{t=0}^{l} [\hat{V}_i^t(X_i^t, \hat{a}_1^t, ..., \hat{a}_{i-1}^t, \hat{a}_{i+1}^t, ..., \hat{a}_N^t;\alpha_i) - (r_i^t + \gamma\hat{V}_i^{t+1}(X_i^{t+1}, \hat{a}_1^{t+1}, ..., \hat{a}_{i-1}^{t+1}, \hat{a}_{i+1}^{t+1}, ..., \hat{a}_N^{t+1};\alpha_i^{\text{old}}))]^2 \tag{7}$$

where $\hat{V}_i^t(X_i^t, \hat{a}_1^t, ..., \hat{a}_{i-1}^t, \hat{a}_{i+1}^t, ..., \hat{a}_N^t;\alpha_i)$ refers to the estimated state value, and $\alpha_i$ represents parameters of critic for agent $i$. $\alpha_i^{\text{old}}$ will copy $\alpha_i$ every multiple training procedures and is utilized to reduce variance along training.

For joint action tracker, it is used to track policies of each agent in the system and therefore relaxes the assumption of other agents' policies. Given a global state at a certain step, joint action tracker infers what each agent will act. Instead of constructing policy estimator for each agent separately (Lowe et al., 2017), in this paper we use a deep neural network to parameterize joint action tracker and train it as a supervising learning problem. The objective is to minimize mean square error (MSE) between the approximated joint action and the actual one:

$$L_{\text{tracker}}(\phi) = \sum_{t=0}^{l} ||\boldsymbol{a_t} - \hat{\boldsymbol{a}}(O^t;\phi)||_2^2 \tag{8}$$

where $\boldsymbol{a_t}$ denotes actual joint action at step $t$, and $\hat{\boldsymbol{a}}(O^t;\phi)$ is the corresponding approximation, and $\phi$ represents parameters of joint action tracker.

Subsequently, we introduce an algorithm to perform optimization, and therefore learn holding control based on MPPO-T. In particular, since agents are homogeneous in the system, the policies can be trained more efficiently with parameters sharing.

**Algorithm 1.** Learning algorithm for MPPO-T

```
Set time horizon $T$ steps for each simulation, batch size $b$ for train joint action tracker.
Initialize memory buffer $M_i = \varnothing$, $i = 1, 2, ..., n$ for $n$ agents, buffer $B = \varnothing$.
Initialize critic and actor parameters $\alpha_i$, $\theta_i$ for each agent, joint action tracker parameters $\phi$.
for ep = 1 to max-episodes do
    for $t$ = 1 to $T$ do
        Obtain state observation $O_t = [X_1^t, ..., X_n^t]$ for joint action tracker,
        Approximate joint action $\hat{\boldsymbol{a}}$ given $O_t$.
        for $i$ = 1 to $n$ do
            if $b_i$ arrives stop then
                Sample action $a_i^t$ using actor neural network, given current state $X_i^t$.
                if current stop is not the first stop $b_i$ visits then
                    Receive reward $r_i^t$.
                    Store tuple $(X_i^{t-1}, X_i^t, r_i^t, a_i^t, \hat{\boldsymbol{a}})$ in $M_i$.
                end if
            end if
            Store tuple $(O_t, \hat{\boldsymbol{a}}, \boldsymbol{a} = [a_1^t, ..., a_n^t])$ in $B$.
            if number of tuples in $B \geqslant b$
                Update $\phi$ with gradient descent on 8.
                Set $B = \varnothing$.
            end if
        end for
    end for
    for $i$ = 1 to $n$ do
        if $M_i \neq \varnothing$ then
            Update $\theta_i$ with gradient ascent on 6.
            Update $\alpha_i$ with gradient descent on 7.
            Set $M_i = \varnothing$.
        end if
    end for
end for
```

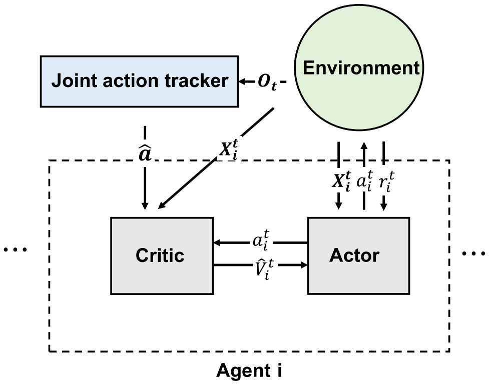


**Fig. 3.** Illustration of MPPO-T.

## 4. Experiments

### 4.1. Simulation setup

We create the simulation environment following the settings presented in Section 2 and Fig. 1. The corridor consists of $m = 12$ uniformly distributed bus stops and a fleet of $n = 6$ buses. All buses have a constant speed $v = 20$ km/h and the travel time excluding dwelling and holding between two consecutive bus stops is 4 min (i.e., the distance between two adjacent stops is 1.33 km). The alighting and boarding times per passenger are set to the same values as in Chen et al. (2016): $t_a = 1.8$ s/pax and $t_b = 3.0$ s/pax. We set the maximum holding time $\Delta T = 180$ s. In addition, due to operation constraints in practice, holding decisions will be ignored if the holding time $\Delta d_i^t$ is less than 30 s.

The simulation starts with 10 passengers waiting at bus each stop and all buses being evenly distributed at every two stops. The passenger arrival dynamics at each stop is governed by a Poisson distribution and the arrival rate for each stop is given by Fig. 4. We set the simulation resolution (one step) as 1 s. The learning procedure is implemented with Python 3.6 and TensorFlow 1.13, and the reinforcement learning framework is embedded into the simulator to capture agent interactions.

### 4.2. Evaluation metrics

The proposed multi-agent holding (MH) strategy will be tested and compared with the following five baseline strategies:

- No-holding control (NH). NH refers to the case where no holding control is introduced proactively. However, since buses are not allowed to overtake others, in the event of bus bunching the bus fallen behind will be held compulsorily.
- Naive hard-holding control (HH). Holding control will be activated whenever the forward headway $h^-$ is smaller than a pre-specified value $H_0$, and the holding time is determined by $d = H_0 - h^-$. Note the planned headway is set to 6 min as in Chen et al. (2016).
- Naive schedule-based holding control (SH). Holding control will be activated whenever a bus arrives earlier according to the timetable. In this case, the bus will hold until the scheduled departure time in the timetable.
- Backward headway-based holding control (BH) (Xuan et al., 2011). The holding time is $d = \beta h^+$, where $h^+$ is the backward headway of arriving bus and $\beta$ is parameter to be set.
- Forward headway-based holding control (FH) (Daganzo, 2009). The holding time is $d = \max\{0, \bar{d} + g(H_0 - h^-)\}$, where $h^-$ is forward headway of the arriving bus, $\bar{d}$ is the average delay at equilibrium, and $g > 0$ is a control parameter. We use the same parameters as in Daganzo (2009).

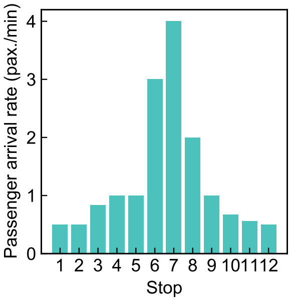
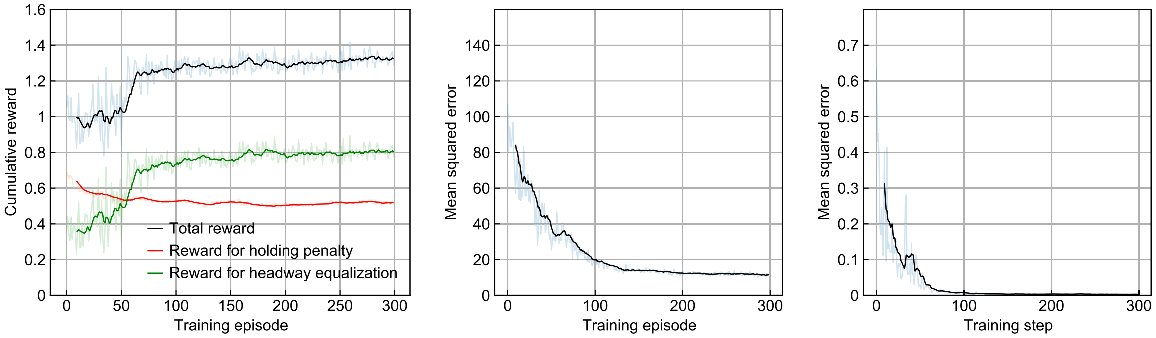


**Fig. 4.** Passenger arrival rate at stops.



(a) Cumulative reward in each episode for the actor neural network.

(b) MSE for the critic neural network.

(c) MSE for the joint action tracker.

**Fig. 5.** Training of the proposed MDRL framework.

In terms of the level of intervention, NH can be considered a special case with no intervention since all holding activities are resulted from the no overtaking constraints; HH and SH are less active since a pre-determined headway and timetable are used as guideline for holding control; BH and FH are more active control in the sense that holding decisions are made based on real-time information for each bus at each bus stop.

The evaluation metrics are chosen to reflect the efficiency and reliability of bus services from both passengers' and operators' perspectives (Berrebi et al., 2018). In this paper, we use the following three evaluation metrics:

- Average holding period, which measures how long the buses have been held during the simulation. This metric characterizes the degree of intervention to the operation, and it is also a proxy to the additional travel time suffered by the on-board passengers.
- Average waiting time per passenger, which calculates how long on average passengers have been waiting at their departure bus stop. Average waiting time is primary indicator to quantify service level and service reliability.
- Occupancy dispersion at each stop. As mentioned, when bus bunching happens, the loading profiles (occupancy) of vehicles will be highly unbalanced with the bus full of passenger while the following buses being empty. Therefore, here we use occupancy dispersion to evaluate service reliability and trip experience. Note the dispersion is computed as variance-to-mean ration (VMR) for the occupancy of all buses when passing a particular bus stop.

### *4.3. Model training*

We set the sizes of the actor neural network and the critic neural network to (3, 256, 1) and (8, 256, 1), respectively, for each agent. The size of the joint action tracker neural network is set to (18, 512, 6). In addition, the hyper-parameter $\epsilon$ in PPO is set to 0.2. We train the model for 300 episodes, and each episode is a 3-h equivalent simulation with 25-min warm-up time. The overall training process takes around one hour on a laptop with Intel Core i7 CPU with 16 GB RAM. The training time can be greatly reduced if using GPU.

Fig. 5 presents the training results over the 300 episodes for the actor neural network, the critic neural network, and the joint-action tracker, respectively. As can be seen, the training process converges after about 150 episodes. The black curve in Fig. 5a shows the variation of total reward (Eq. (3)) over the whole training process, and the green and red curves show the values of $\exp(-|h_i^{t-} - h_i^{t+}|)$ (reward for headway equalization) and $w\exp(-a_i^t)$ (reward for holding penalty), respectively. We can see that the reward for headway equalization plays a more important role in policy training. However, it should be noted that the holding penalty term also plays a critical role in preventing over-intervention.

### *4.4. Results and analysis*

Considering the uncertainty of bus bunching, we conduct simulations with different horizons ranging from 2 h to 8 h. In addition, each evaluation will conduct 20 experiments to evaluate model reliability. Before we start, we first analyze different control strategies on a microscopic level. Fig. 6 presents the bus trajectories in our simulation when no control/holding strategies are implemented (NH). This figure shows that, without any interventions, a circle bus is in fact very vulnerable to bus bunching problems.

Fig. 7 presents the bus trajectories for the proposed MH strategy together with other baseline models. The black dots on top of the trajectories show the total holding control time triggered at each bus stop, and the deeper color indicates a longer holding time. As can be seen in Figs. 7(a) and 7(b), HH and SH only start imposing control after severe bus bunching happens. As a result, they can not effectively avoid bus bunching due to the considerable uncertainties in the system. In contrast, those more active strategies (BH, FH and MH) are more effective in preventing bus bunching from happening. Among these three strategies, MH clearly imposes less holding control than the others, in particular at stops $s_6$ and $s_7$ with larger passenger demand. This is mainly because the proposed MH can effectively take into account the number of waiting passengers as part of the state observation, while others do not take full advantage of this information, which results in unnecessary intervention/holding decision in the system. In addition, since less

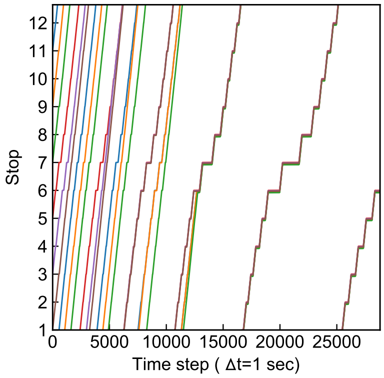
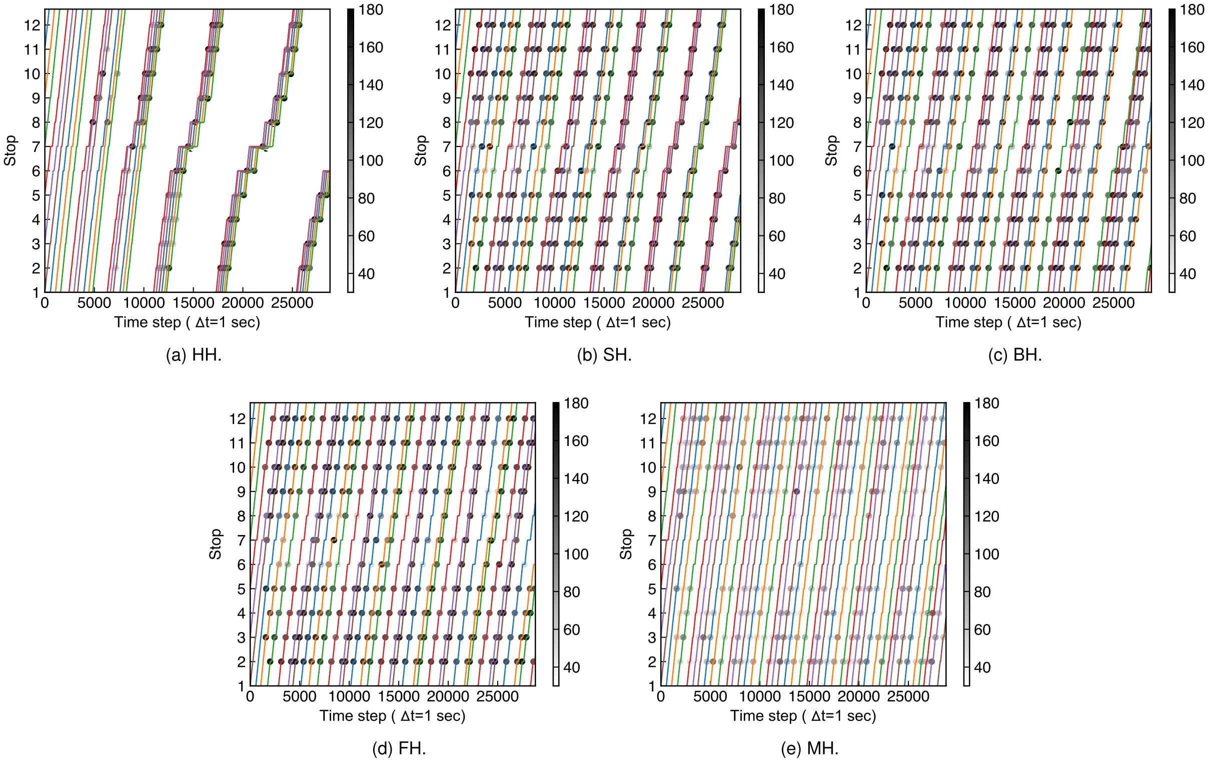


**Fig. 6.** Bus trajectories under NH.



(a) HH.

(b) SH.

(c) BH.

(d) FH.

(e) MH.

**Fig. 7.** Bus trajectories under different holding strategies.

holding time corresponds less overall travel time on the whole bus route, the fleet control by MH are used more efficiently with an increased number of services over the whole simulation period.

Fig. 8 shows corresponding total loads for each bus at different stops over the entire operating horizon. Intuitively, when there is no control (i.e., NH), we observe obvious uneven bus loads, which means a waste of public transport resource. Notably, HH and SH struggle to improve this situation. In contrast, BH, FH and MH present less variability in the bus loads due to their efficiency in avoiding bus bunching. The proposed MH has the best occupancy dispersion performance. In addition, we find that the assumption of unlimited bus capacity does not affect our results, as the proposed MH can effectively address both the bus bunching and load imbalance issues.

Apart from the micro-scale inspection, we conduct further evaluation on a macroscopic level (i.e., transits system). As shown in Fig. 9, the indexes mentioned above are utilized for comparison. We compare the indexes by averagings results from 20 simulations, and the error bar represents the double standard variance for each single time period (i.e., mean +/− 2*standard variance). Specifically, the left panel presents the average holding period in the system for the entire operation horizon. At the beginning, since buses are emitted in even headway, there is less holding control, especially for the strategies that only activate control when bus bunching is closed to come (i.e., NH, HH, SH). However, as time goes by, holding control based on NH, HH and SH increases

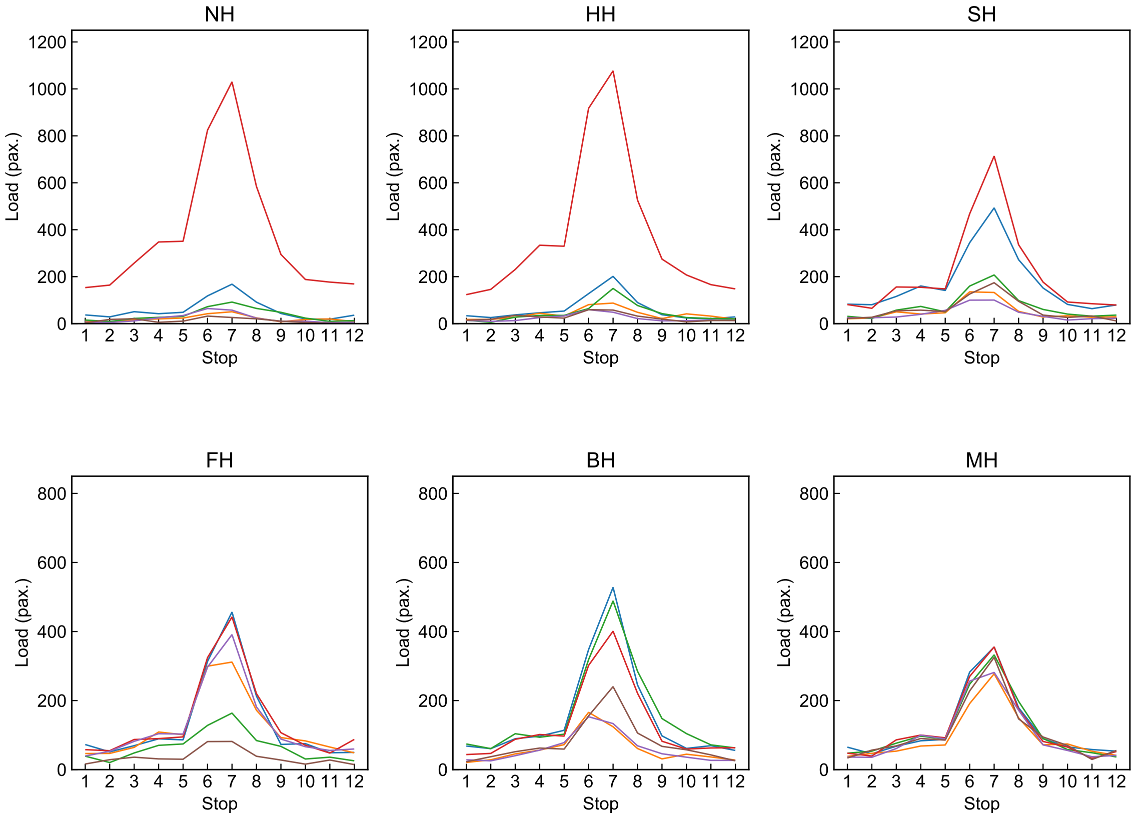
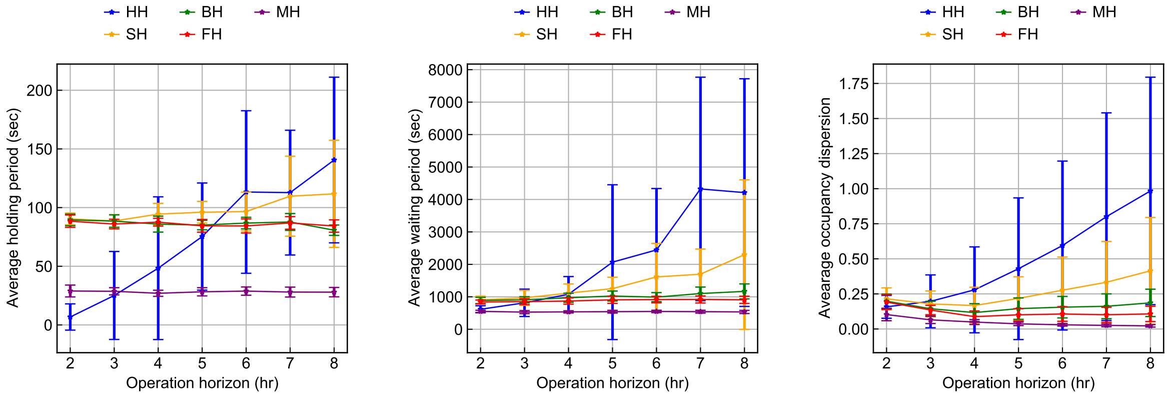


**Fig. 8.** Bus loads at different stops under different strategies.



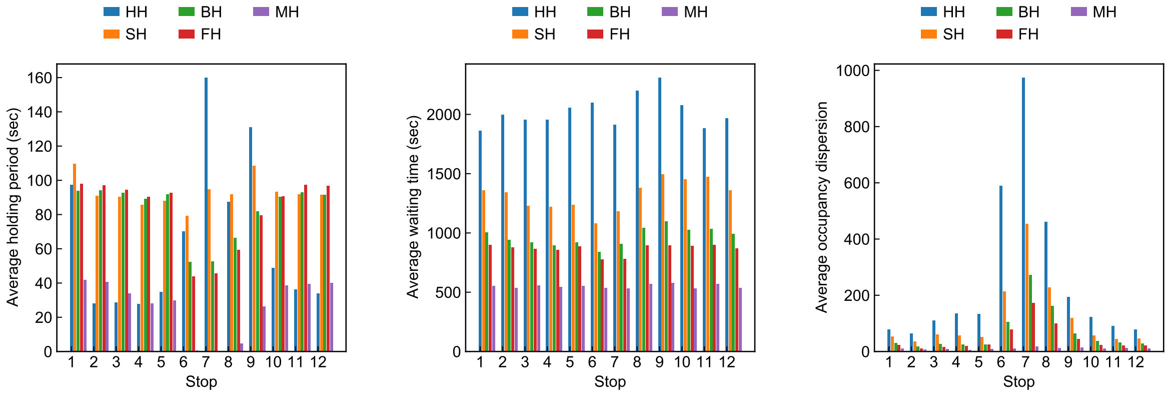
(a) Average holding period.

(b) Average waiting time.

(c) Average occupancy dispersion.

**Fig. 9.** Comparison of three indexes along operation horizon in view of system..

significantly. In contrast, headway based control (i.e., BH, FH) and the proposed method keep a stable control along the operation. The proposed method (MH) imposes the least intervention. This is due to the fact that MH considers additional passenger flow information, and coordination among agents can help avoid unnecessary control. In fact, the holding control should be activated in advance, once the system runs without proactive holding, the buses are more likely to be trapped into bunching state and the situation will be exacerbated as the vicious circle comes into effect. The middle panel analyzes the average waiting time per passenger. Interestingly, the waiting time seems to be stable along the operation horizon for FH, BH and MH, while continuously increases for other strategies. The reason is that FH, BH and MH present an effective control strategy and avoid bus bunching, therefore maintain a more stable headway than others. Otherwise, unstable headway can have lasting effects on the system and on passenger experience (Berrebi et al., 2018). In addition, the proposed method outperforms others since there is no pre-specified headway or schedule. In other words, the agents (i.e., buses) are required to explore an effective headway heuristically by maximizing the proposed cumulative reward on their own. The right panel shows average occupancy dispersion among the system. The



(a) Average holding time.

(b) Average waiting time.

(c) Average occupancy dispersion.

**Fig. 10.** Comparison of three indexes in each stop for entire operation horizon.

longer the system operates, the larger probability that the bus bunching occurs. Similar to the average waiting time, FH, BH and MH still present higher stability and less occupancy dispersion. Since bus bunching results in unequal headway among the system, the number of passengers served by different buses at each stop varies greatly: whilst some buses may be full of passengers, others are totally empty. Consequently, an effective strategy should have a control on headway that can lead to more balanced transit resource allocation.

Fig. 10 presents the comparison of different control strategies on the entire bus route in detail. As we can see, the proposed MH strategy clearly outperforms other strategies in terms of all the three evaluation metrics, preventing bus bunching with a minimum level of intervention. Interestingly, we can see that the less active holding control strategies (i.e., HH, SH) in fact require longer holding time in the long run. Moreover, due to the bad performance, passenger waiting times for HH and SH are also much longer than other strategies. This again indicates that proper holding is important to maintaining system efficiency in bus operation. The shortest error bar for MH demonstrates that MH can prevent bus bunching in each single simulation, thus the evaluation indexes are much more stable. Our results further suggest that the global coordination among agents and the long-term reward are essential in designing more efficient and reliable control strategies. Moreover, by excluding a specific headway/schedule in the reward function, we actually provide more flexibility and freedom to the learning algorithm, which in turn allows the agents to explore a more effective policy by themselves for the whole system.

Overall, the proposed MH strategy effectively addresses the bus bunching problem without introducing strong interventions, and in the meanwhile it is also beneficial in terms of reducing passenger waiting time and travel time, balancing the level of crowdedness in different vehicles, and offering more frequent and reliable services. To further verify the robustness of our model, we conduct a new set of experiments by relaxing the simulation setup of uniformly distributed bus stops. In these scenarios, we assign the bus stops randomly, and we find that all the results hold in these new scenarios.

## 5. Conclusions and future work

In this paper, we propose a multi-agent deep reinforcement learning framework to develop dynamic holding control strategies to address the bus bunching problem. By carefully designing the reward function, the agents (i.e., buses) can effectively learn global holding strategies by pursuing headway equalization and avoiding heavy intervention in the meanwhile. To achieve system-wide fleet coordination among all agents, we propose to use a joint-action tracker on top of the basic actor-critic framework when exploring the near-optimal holding strategies. To train the deep neural network, we also design an efficient learning algorithm using proximal policy optimization. Finally, we demonstrate the superiority of the proposed multi-agent deep reinforcement learning framework through extensive experiments against several baseline models.

For future work, we would like to extend this framework to further account for the uncertainties in fleet operation, such as the randomness of traffic signal control, the diversity in driving behavior, and certain extreme scenarios in fleet operation (e.g., vehicle break-down). This will also allow us to better validate our models in real-world applications and design more robust solutions. In addition, with the increasing penetration of mobile transit applications (e.g., https://transitapp.com/), real-time passenger demand information can be obtained at a global scale for all bus stop. This information can be further integrated into the reinforcement learning framework to achieve better performance. Finally, we would like to work with transit agencies to test the proposed framework in real-world fleet operations.

## CRediT authorship contribution statement

**Jiawei Wang:** Conceptualization, Methodology, Formal analysis, Writing - original draft. **Lijun Sun:** Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision, Funding acquisition.

## Acknowledgment

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Mitacs Canada, and the Canada Foundation for Innovation (CFI).

## References

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

Aslani, M., Mesgari, M.S., Wiering, M., 2017. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. Transp. Res. Part C: Emerg. Technol. 85, 732–752.

Bartholdi III, J.J., Eisenstein, D.D., 2012. A self-coördinating bus route to resist bus bunching. Transp. Res. Part B: Methodol. 46 (4), 481–491.

Berrebi, S.J., Hans, E., Chiabaut, N., Laval, J.A., Leclercq, L., Watkins, K.E., 2018. Comparing bus holding methods with and without real-time predictions. Transp. Res. Part C: Emerg. Technol. 87, 197–211.

Cats, O., Larijani, A.N., Koutsopoulos, H.N., Burghout, W., 2011. Impacts of holding control strategies on transit performance: Bus simulation model analysis. Transp. Res. Rec. 2216 (1), 51–58.

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

Daganzo, C.F., 2009. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. Transp. Res. Part B: Methodol. 43 (10), 913–921.

Daganzo, C.F., Pilachowski, J., 2011. Reducing bunching with bus-to-bus cooperation. Transp. Res. Part B: Methodol. 45 (1), 267–277.

Delgado, F., Munoz, J.C., Giesen, R., 2012. How much can holding and/or limiting boarding improve transit performance? Transp. Res. Part B: Methodol. 46 (9), 1202–1217.

Delgado, F., Munoz, J.C., Giesen, R., Cipriano, A., 2009. Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. Transp. Res. Rec. 2090 (1), 59–67.

Eberlein, X.J., Wilson, N.H., Bernstein, D., 2001. The holding problem with real–time information available. Transp. Sci. 35 (1), 1–18.

El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto. IEEE Trans. Intell. Transp. Syst. 14 (3), 1140–1150.

Estrada, M., Mensión, J., Aymamí, J.M., Torres, L., 2016. Bus control strategies in corridors with signalized intersections. Transp. Res. Part C: Emerg. Technol. 71, 500–520.

Fu, L., Liu, Q., Calamai, P., 2003. Real-time optimization model for dynamic scheduling of transit operations. Transp. Res. Rec. 1857 (1), 48–55.

Kuyer, L., Whiteson, S., Bakker, B., Vlassis, N., 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 656–671.

Li, L., Lv, Y., Wang, F.-Y., 2016. Traffic signal timing via deep reinforcement learning. IEEE/CAA J. Automatica Sin. 3 (3), 247–254.

Littman, M.L., 1994. Markov games as a framework for multi-agent reinforcement learning. Mach. Learn. Proc. 1994, 157–163.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems. pp. 6379–6390.

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

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

Newell, G.F., Potts, R.B., 1964. Maintaining a bus schedule. In: Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne, Australia, vol. 2.

Petit, A., Ouyang, Y., Lei, C., 2018. Dynamic bus substitution strategy for bunching intervention. Transp. Res. Part B: Methodol. 115, 1–16.

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

Sun, A., Hickman, M., 2005. The real–time stop–skipping problem. J. Intell. Transp. Syst. 9 (2), 91–109.

Sutton, R.S., Barto, A.G., et al., 1998. Introduction to Reinforcement Learning, vol. 2 MIT Press Cambridge.

Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y., 2000. Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing systems. pp. 1057–1063.

Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., Bayen, A.M., 2017a. Flow: Architecture and benchmarking for reinforcement learning in traffic control. arXiv preprint arXiv:1710.05465.

Wu, W., Liu, R., Jin, W., 2017b. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. Transp. Res. Part B: Methodol. 104, 175–197.

Xuan, Y., Argote, J., Daganzo, C.F., 2011. Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis. Transp. Res. Part B: Methodol. 45 (10), 1831–1845.

Zhao, J., Dessouky, M., Bukkapatnam, S., 2006. Optimal slack time for schedule-based transit operations. Transp. Sci. 40 (4), 529–539.

Zhao, S., Lu, C., Liang, S., Liu, H., 2016. A self-adjusting method to resist bus bunching based on boarding limits. Math. Problems Eng.