Robust Dynamic Bus Control: A Distributional Multi-Agent Reinforcement Learning Approach

Jiawei Wang and Lijun Sun, Senior Member, IEEE

Abstract—The bus system is a critical component of sustainable urban transportation. However, the operation of a bus fleet is unstable in nature, and bus bunching has become a common phenomenon that undermines the efficiency and reliability of bus systems. Recently, research has demonstrated the promising application of multi-agent reinforcement learning (MARL) to achieve efficient vehicle holding control to avoid bus bunching. However, existing studies essentially overlook the robustness issue resulting from perturbations and anomalies in a transit system, which is of utmost importance when transferring the models for real-world deployment/application. In this study, we integrate implicit quantile network and meta-learning to develop a distributional MARL framework—IQNC-M—to learn continuous control. The proposed IQNC-M framework achieves efficient and reliable control decisions through better handling various uncertainties in real-time transit operations. Specifically, we introduce an interpretable meta-learning module to incorporate global information into the distributional MARL framework, which is an effective solution to circumvent the credit assignment issue in the transit system. In addition, we design a specific learning procedure to train each agent within the framework to pursue a robust control policy. We develop simulation environments based on real-world bus services and passenger demand data and evaluate the proposed framework against both traditional holding control models and state-of-the-art MARL models. Our results show that the proposed IQNC-M framework can effectively handle both efficiency and reliability of the transit system.

Index Terms—Bus bunching, robust holding control, multi-agent reinforcement learning, distributional reinforcement learning, meta-learning.

I. INTRODUCTION

The operation of bus systems is unstable due to great uncertainties in traffic states and passenger demand. Bus bunching has been a long-standing operation problem resulting from these uncertainties. If a bus is slightly delayed on the route, it will be more likely to encounter more waiting passengers at the next bus stop, and then the dwell time serving boarding/alighting passengers will also increase. As a result, a delayed bus will be further delayed on a route. Due to the instability of the system, bus bunching has become the most critical operational issue that undermines the reliability and efficiency of bus services. As a result, real-time bus fleet control is requisite for improving the performance of bus systems. With respect to bus fleet control, transit system is essentially the feedback control system, where sensors, control algorithms, and controllers are arranged in each control loop to ensure operational regularity and efficiency. Specifically, the sensors of the system can be any manufacture that can observe the transit dynamic, such as GPS reporting the location and speed of the bus fleet. Such observation from sensor will be fed into specific control algorithms for strategy calculation. Then the strategy will intrigue the controllers (i.e., buses or bus stops) to implement actions such as holding control, and boarding limit [1], [2], [3]. With recent advances in artificial intelligence [4], [5], researchers have identified reinforcement learning (RL) as a promising solution to solve complex traffic control problems, such as traffic signal control [6], [7] and ramp metering control [8]. To address the bus bunching problem, recent studies have also modeled bus holding control of a bus fleet in RL and multi-agent reinforcement learning (MARL) framework [see e.g., [9], [10], [11]]. However, modeling bus holding control in MARL is a challenging problem. First, some common and tricky challenges exist in learning robust control policies in MARL, such as the non-stationary dynamics and the complex credit assignment issue. Second, bus holding is an asynchronous control problem—a bus can only implement the holding action when it arrives at a bus stop. As a result, we cannot directly adapt existing MARL frameworks to model bus fleet operation. To address the asynchronous issue, Wang and Sun [12] propose an inductive critic to improve credit assignment among agents in an asynchronous setting, and the model has demonstrated superior performance in learning bus holding control policies to avoid bus bunching. Despite the promising results, there still exist several critical challenges in adapting the RL for real-world operations/deployments. On the one hand, elevating the efficiency and stability of MARL is still tricky for the large multi-agent system. Specifically, as a critical problem of MARL, the credit assignment may not require an explicit formulation [13], since approximating the contribution of each agent may introduce an approximation error that hinders policy training. On the other hand, how to achieve and examine robust control are lack of a mature framework. Previous RL-based fleet control methods have not fully examined the robustness of the solutions [9], [11], [12], [14],
which is of utmost importance for real-world applications. As a result, these control policies may behave less efficiently under unexpected scenarios where the traffic state and transit demand are uncertain, such as demand impulse at some stops (i.e., there is a special event in the neighborhood) and incidents causing traffic congestion. To fill these gaps, we propose a distributional MARL framework—Implicit Quantile Network for Continuous control with Meta-learning (IQNC-M)—to learn robust bus holding control policies. In particular, we introduce an interpretable meta-learning module to allow efficient distributional MARL, meanwhile addressing the aforementioned non-stationary dynamics and complex credit assignment issues when modeling bus fleet as a multi-agent system. Our main contributions can be summarized as follows:

- To the best of our knowledge, this study is the first to introduce distributional MARL to achieve robustness in bus fleet operations and holding controls toward uncertain transit demand and traffic state.
- We design a meta-learning-based framework to adapt distributional RL for efficient multi-agent control policy learning.
- We design a training procedure for learning robust holding control policies in the distributional MARL framework.
- We evaluate the performance—in terms of both effectiveness and robustness—of the proposed framework through extensive simulation experiments on four real-world bus routes.

The rest of the paper is organized as follows. Section II presents the bus fleet holding control problem and discusses related works on traditional optimization-based models and recent RL- and MARL-based solutions. Section III introduces the background and preliminary knowledge of this study. Section IV presents the proposed distributional MARL approach to model bus fleet control. Experimental studies are presented in Section V to evaluate the proposed model. Finally, Section VI summarizes some concluding remarks and discusses future research directions.

II. RELATED WORK

A. Research on Bus Fleet Control Strategy

Bus holding control has long been considered the most effective solution to reduce bus bunching because it can make a good compromise among control efficiency, implementation practicability, and service satisfaction of transit users [1], [11], [15], [16]. According to the previous studies, there are mainly two traditional approaches to developing holding control strategies with respect to the solution method. The first is the optimization-based approach [1], [17], [18], which mainly focuses on deriving holding period/duration on a rolling horizon analytically by optimizing objectives related to system efficiency such as passenger waiting time. These studies need to model the future states of the system and implement optimization at each decision stage, which turns out to be less efficient. The second approach is to develop rule-based holding control strategies based on the linear control law [2], [19], [20]. These studies are computationally efficient but may be less reliable due to the simple linear assumptions in reproducing the complex dynamics in transit operations. Notably, most recent studies have identified that incorporating real-time information is a key solution to efficient transit control. For example, Berrebi et al. [21] evaluated the holding control methods and concluded that predictive control based on real-time information could achieve the best compromise between headway regularity and holding cost. Gkiotsalitis and Van Berkum [3] introduced an analytic solution for the single variable bus holding problem that considers the real-time passenger demand and vehicle capacity limits. Wang and Sun [11] designed MARL bus fleet control framework considering real-time global control decisions. These studies demonstrate that real-time information can offer more efficient and more effective control policies.

Uncertainties in transit demand (i.e., passenger arrival rate) and traffic state (i.e., on-road running time) are the two most important factors that undermine the robustness and reliability of transit operation. Recent studies have taken these uncertainties into account in building analytical models. For example, Xuan et al. [2] considered stochastic on-road running time in developing robust control holding strategies. Wu et al. [16] proposed adaptive control strategies that take into account the dynamic redistribution of passenger queues. van der Werff et al. [22] developed an assessment framework to systematically assess the quality of a transit service under headway-based holding strategies. Notably, they evaluate the control performance with two designed scenarios: trip detour and traffic disruption. Li et al. [23] derived robust control strategies for schedule adherence and headway regularity considering both the randomness in demand and traffic state. Although these studies have contributed to developing robust bus control, there are still some major limitations. On the one hand, most previous studies only introduce perturbations in demand and speed as a way to incorporate uncertainties; however, these scenarios are essentially easy to solve and should be categorized as normal scenarios in daily transit operations. A more challenging task is actually to adapt for extreme scenarios and anomalous events, such as on-road incidents and demand surges, which may lead to cascading failures in the system. In addition, existing models cannot effectively learn from the past to update the control policy, as they often require an exact mathematical model of the complex transit system with a prior setting. Therefore, they become less efficient in handling the unstable dynamic of the transit system.

B. Research on Transit Control With Reinforcement Learning

RL provides an end-to-end control learning paradigm over traditional methods, allowing model-free control and continuous learning with long-term consideration [24]. Thanks to the growing computation ability, researchers have extended RL to deep reinforcement learning (DRL) by exploiting deep neural networks [4], [25]. Furthermore, MARL, which is the DRL scheme for the multi-agent case, provides an efficient framework for exploring a cooperative control solution. Recent years have witnessed increasing research on DRL contributing
to the transit control problem. For example, Alesiani and Gkiotsalitis [10] implemented holistic holding control based on DRL. Chen et al. [9] and [11] utilized MARL framework to achieve decision coordination among buses in the transit system. Wang and Sun [12] proposed a multi-agent deep reinforcement learning framework to achieve efficient holding control by addressing the asynchronous issues in transit operation. All these studies have demonstrated the superior performance of RL on vehicle holding control. Nevertheless, they mainly focus on elevating control efficiency in daily normal bus fleet operation. Specifically, their training and test procedure are based on normal operation (i.e., there is less uncertainty in traffic state and transit demand). As a result, these studies can not demonstrate the robustness issue of MARL in transit control. Therefore, it remains an open question how to achieve robust control under uncertainty and extreme scenarios with RL based control scheme.

III. BACKGROUND
A. Reinforcement Learning Based Control

RL provides an adaptive scheme to explore optimal control policy in a system that is governed by a Markov decision process (MDP). Specifically, the learner and controller in this system is red to as the agent, while all the other components are regarded as the environment [24]. At each decision step $t$, the agent decides an action $a_t \in \mathcal{A}$ based on its current policy $\pi_t$, given a state $s_t \in \mathcal{S}$ observed from the environment. Then, the agent moves to the next state $s_{t+1} \in \mathcal{S}$, and receives reward signal $r_t \in \mathcal{R}$ as a result of this interaction. Based on the MDP assumption, the dynamic of this process can be represented by the transition probability: $p(s', r | s, a) \equiv \Pr \{s_{t+1} = s', r_t = r | s_t = s, a_t = a\}$.

In the RL task, the goal of the agent is to learn an optimal policy $\pi^*$ to maximize the cumulative reward signal along the control horizon $H$. Generally, the cumulative reward signal can be formulated as the expected return with a discounted factor $\gamma \in [0, 1]$:

$$R_t = \sum_{k=0}^{H} \gamma^k r_{t+k}. \quad (1)$$

There are mainly two categories of RL methods—value-based methods and policy-based methods. Value-based methods, such as Q-learning and Deep Q-Networks (DQN) [4], use state-action value function $Q(s, a)$ for policy evaluation. The basic idea is that after achieving a good estimate of the optimal state-action value, $Q^*(s, a)$, the optimal policy $\pi^*$ will be the one choosing action $a$ with the highest state-action value at state $s$. Most RL methods estimate the state-action value function by using the Bellman equation as an iterative update [26]. This scheme is also red to as bootstrapping [24], i.e., updating the state-action value estimate for a state based on the estimated values $\overline{Q}$ of subsequent states:

$$Q^\pi(s_t, a_t) = E_{s_{t+1}} \left[ r_t + \gamma \max_{a_{t+1}} \overline{Q}(s_{t+1}, a_{t+1}) \right]. \quad (2)$$

As an extension of Q-learning, DQN introduces deep neural network to learn/approximate $Q^\pi$ by minimizing the following loss function:

$$\mathcal{L} (\theta) = E_{s_t,a_t,r,s'} \left[ Q^\pi(s_t, a_t) - \left( r_t + \gamma \max_{a'} \overline{Q}(s', a') \right) \right]. \quad (3)$$

Policy-based methods, such as policy gradient (PG) [27] and deterministic policy gradient (DPG) [28], parameterize policy as $\pi_\theta$ and then directly optimize $\pi_\theta$ to achieve a larger cumulative reward. Formally, the objective of these methods is to maximize the cumulative reward $J(\theta) = E_{s_t,a_t} [R_t]$. In PG, the action is sampled from a parametric probability distribution $a_t \sim \pi_\theta(a_t | s_t) = P(a_t | s_t, \theta)$. The policy parameters are updated through gradient ascent with respect to the cumulative reward:

$$\nabla_\theta J(\theta) = E_{s_t,a_t} \left[ \nabla_\theta \log \pi_\theta(a_t | s_t) Q^\pi(s_t, a_t) \right]. \quad (4)$$

As PG adopts a stochastic policy, it suffers from large variances during the learning process. To address this issue, DPG suggests a deterministic mapping from state to action: $a_t = \pi_\theta(s_t)$. By doing so, we only need to sample from the state space to update the parameters, and the gradients can be calculated as:

$$\nabla_\theta J(\theta) = E_{s_t} \left[ \nabla_\theta \pi_\theta(a_t | s_t) Q^\pi(s_t, a_t) \right]. \quad (5)$$

In general, the cumulative reward $R_t$ can be estimated by learning the state-action value function $Q(s, a)$. Such a framework is red to as actor-critic [24], in which the state-action value estimator is called the critic, and the policy is modeled on the actor.

B. Distributional Reinforcement Learning

Different from traditional RL methods that learn the expectation of the cumulative reward or the state-action value, distributional RL explores the state-action value distribution and thus allows us to account for the intrinsic uncertainty of an MDP. In distributional RL, the state-action value distribution can be modeled as a distributional Bellman equation [29]:

$$Z(s_t, a_t) \doteq r(s_t, a_t) + \gamma Z(s_{t+1}, a_{t+1}), \quad (6)$$

where the state-action value distribution $Z(s_t, a_t)$ takes into account the uncertainty from reward signal $r(s_t, a_t)$, the transition to the next state $(s_{t+1}, a_{t+1})$, as well as the next random return $Z(s_{t+1}, a_{t+1})$.

To facilitate state-action value distribution learning within traditional DRL frameworks (e.g., DQN), a state-of-the-art solution is to approximate the target distribution with quantile function and approximate the state-action value [30] by:

$$\hat{Q}(s_t, a_t) = E_{\tau \sim \mathcal{U}[0, 1]} [Z_{\tau}(s_t, a_t)] \approx \sum_{k=0}^{K-1} (\tau_{k+1} - \tau_k) Z_{\tau_k}(s_t, a_t), \quad (7)$$

where $\tau_k = \frac{k+1+\tau}{K+1}$ and $Z_{\tau}(s_t, a_t)$ represents the quantile function at fraction $\tau \in [0, 1]$ for $Z(s_t, a_t)$. Then, we can sample $\tau$ from a uniform distribution $\mathcal{U}[0, 1]$ to produce a state-action value sample of $Z(s_t, a_t)$.

According to previous studies [29], [30], distributional RL exhibits superior performance in terms of both training
convergence and execution performance. By modeling the value distribution, we can better characterize the intrinsic uncertainties in MDP and thus identify more robust and safer control policies [31].

C. Distorted State-Action Value Distribution

The distorted expectation can be expressed as a weighted average of quantiles [32]. Specifically, given a distortion function \( g : [0, 1] \rightarrow [0, 1] \) satisfying \( g(0) = 0 \) and \( g(1) = 1 \), the distorted expectation of state-action value distribution \( Z \) is \( \int_{0}^{1} Z_{t}dg(\tau) = \int_{0}^{1} g'(\tau)Z_{t}d\tau \), where \( g'(\tau) \) is the derivative of \( g \) and it can be considered the distortion weights for quantiles. For example, Wang [33] proposed the following distortion function:

\[
g(\tau) = \Phi(\Phi^{-1}(\tau) + \beta),
\]

(8)

where \( \Phi \) represents the cumulative distribution function (CDF) of standard normal distribution, and \( \Phi^{-1} \) is the inverse CDF; and \( \beta > 0 \) corresponds to being risk-averse while \( \beta < 0 \) indicates being risk-seeking. Generally, the distorted state-action value expectation can be approximated by:

\[
\hat{Q}(s_t, a_t) = \sum_{k=0}^{K-1} (\tau_{k+1} - \tau_k) g'(\tau_k) Z_{\tau_k}(s_t, a_t).
\]

(9)

As shown in Fig. 1, the distortion weight essentially corresponds to the preference for policies by distorting state-action value distribution. For example, the risk-averse policy generally places lower weight on larger quantiles.

D. Meta Reinforcement Learning

Meta RL introduces meta-learning to improve learning performance [34] and make it easier for agents to generalize to new tasks [35]. Specifically, meta-gradient RL [34] is one of the most popular schemes for optimizing hyper-parameters in RL. In general, the RL parameters \( \theta \) (i.e., parameters of the state-action value approximator or the policy) are updated as follows:

\[
\theta' = \theta + f(d, \theta, \eta),
\]

\[
f = \frac{\partial J}{\partial \theta'},
\]

(10)

where \( J \) is the objective function (e.g., Eq. (3)) for the RL agent and \( \eta \) represents the hyper-parameter for meta learning.

We sample trajectories \( d \) (e.g., \( (s, a, r, s') \)) to calculate gradient for \( J \). In meta-gradient RL, \( \eta \) is updated based on the gradient from the derivative of a differentiable meta-objective \( J' \). To be specific, the derivative is calculated with another sample trajectory \( d' \) and a fixed \( \eta' \) as a preference value:

\[
\frac{\partial J'}{\partial \eta'} = \frac{\partial J'}{\partial \theta'} \frac{\partial \theta'}{\partial \eta'}.
\]

(11)

Intuitively, after every update on \( \theta \), meta RL adapts the meta-parameters \( \eta \) to the direction that achieves better performance.

IV. METHODOLOGY

A. Bus Control Framework

Bus holding control is the most widely used strategy to maintain consistent headway and avoid bus bunching in daily operation [15]. The essential idea of holding control is to let a bus stay longer (i.e., adding a slack period) at a bus stop in addition to the dwell time for passengers to board/alight. Notably, we model bus holding as a control policy in MARL with the objective of improving operational efficiency. The symbols and notations used to describe the bus control framework are provided in Table I.

As shown in Fig. 2, all buses in the transit system follow a bus route. When bus \( b_i \) arrives at a bus stop, we denote by \( h_{i+1} \) the backward headway (time for the following bus \( h_{i+1} \) to reach the current location of \( b_i \)) and \( h_i \) the forward headway (time for \( b_i \) to reach the current location of \( h_{i-1} \)). The bus has access to the number of on-board passengers and it also observes the number of waiting passengers at the bus stop. Each bus will determine holding period/duration once it arrives at a stop (i.e., the green circle) and receive feedback when arriving at the next stop (i.e., the red circle).

We sample trajectories \( d \) and \( s' \) to calculate gradient for \( J \). In meta-gradient RL, \( \eta \) is updated based on the gradient from the derivative of a differentiable meta-objective \( J' \). To be specific, the derivative is calculated with another sample trajectory \( d' \) and a fixed \( \eta' \) as a preference value:

\[
\frac{\partial J'}{\partial \eta'} = \frac{\partial J'}{\partial \theta'} \frac{\partial \theta'}{\partial \eta'}.
\]

(11)

Intuitively, after every update on \( \theta \), meta RL adapts the meta-parameters \( \eta \) to the direction that achieves better performance.
an action (i.e., determine holding period) once it arrives at a bus stop and receives feedback when arriving at the next bus stop. In particular, the authors model bus holding control as an asynchronous control task, in which it is not necessary for all agents to act simultaneously. Following the same framework as in [12], we consider a partially observable scenario in which each bus asynchronously determines the holding duration based on local observations. For a transit system with \( N \) buses, each bus maintains its local state transition: \( \tilde{P}_i : S_i \times \tilde{A} \mapsto S_i, S_i \subseteq S, \tilde{A} \subseteq A \), where \( S_i \) is the observation space of agent \( i \), \( S = \{S_1, \ldots, S_N\} \) and \( A = \{A_1, \ldots, A_N\} \) denote the global state space and the joint action space, respectively. The parametrized policy \( \pi_\theta \) chooses an action \( a_i \) based on local observation \( s_i \), i.e., \( a_i = \pi_\theta (s_i) \). Finally, the reward function for each agent can also be defined independently: \( R = \{R_1, \ldots, R_N\} \). In the following, we define the elements in the RL pipeline for bus fleet operation on a single route. We to each “bus” as an “agent” and use the two terms interchangeably.

1) State: For each agent, we use its local observation of the system as the state. For bus \( b_i \) arriving at bus stop \( k_j \) at time \( t \), we denote its state by \( S_{ij} \). It should be noted that the definition of state plays an important role in the overall RL framework. In order to support reliable and robust decision making, the elements of the state should be sufficiently rich in characterizing the complex dynamics of the system. In the meanwhile, the state should be accessible—something that can be collected from available sensors on a bus—for real-world and real-time implementations. Specifically, we consider two components in defining states for a bus \( b_i \): the first component consists of the forward headway \( h_i \) and the backward headway \( h_{i+1} \); the second component focuses on passenger demand, and we use the number of onboard passengers and the number of waiting passengers at the bus stop to define it. In practice, the first component of vehicle information can be measured in real-time through the automatic vehicle location (AVL) system, and the second component on passenger demand can be obtained from the automatic passenger counting (APC) system and the automated fare collection (AFC) system.

2) Action: Following [11], we model holding time as:

\[
\Delta d_i^t = a_{i,t} \Delta T,
\]

where we consider \( a_{i,t} \) the action of bus \( b_i \) when arriving at a bus stop at time \( t \). Specifically, \( a_{i,t} \in [0, 1] \) is regarded as a strength parameter. Here, \( \Delta T \) is used to restrict the maximum holding duration and avoid over-intervention, and we model \( a_{i,t} \) as a continuous variable to explore the near-optimal policy in an adequate action space. No holding control is implemented when \( a_{i,t} = 0 \).

3) Reward: Although holding control can reduce the variance of bus headway and therefore promote system stability, in the meanwhile, it also introduces additional slack time at stops. As a result, passenger travel time and route operation time (i.e., the total duration from leaving the departure terminal to arriving at the final destination) will also increase, imposing an additional penalty on passenger travel cost and also requiring a larger fleet to maintain service frequency. To balance system stability and operation efficiency, we design the reward function associated with bus \( b_j \) at time \( t \) as:

\[
r_i^t = -(1 - \omega) \times CV^2 - \omega \times a_{i,t},
\]

where \( CV^2 = \frac{\text{Var}[h]}{\mathbb{E}[h]} \) is an indicator for variability of headway \( h \) [37]. Essentially, \( \mathbb{E}[h] \) is almost a constant based on the schedule, and thus \( CV \) is mainly determined by \( \text{Var}[h] \): a small \( CV \) indicates consistent headway values on the bus route, and a large \( CV \) suggests heavy bus bunching. The second term in Eq. (13) penalizes holding decisions and prevents the learning algorithm from making excessive control decisions. Overall, the goal of this reward function is to achieve system stability with as few interventions as possible, with \( \omega \) being a balancing parameter. We set \( \omega = 0.2 \) in this study.

B. Distorted Distributional Multi-Agent Reinforcement Learning for Robust Transit Control

Modeling uncertainty of transit operation using probability distribution is one of the most common solutions to derive robust control policies [38], [39]. While distributional RL provides an efficient framework to consider uncertainty by learning state-action value distribution, there still exist two critical challenges in exploiting distributional RL in a multi-agent transit system. The first is the non-stationary environment as a result of the unstable dynamics of the transit system and the undetermined behavior of each bus; the second is the credit assignment resulting from the undetermined contribution of each agent’s decisions during real-time operation. Motivated by the risk-sensitive RL, which uses the distorted state-action value distribution to learn policies with risk preference [40], we manage to address the above challenges by adapting this idea to distributional MARL. The actor-critic [24] is adopted as basic architecture. Specifically, each agent will maintain an actor network to model policy and a critic network for policy evaluation. As shown in Fig. 3, we consider a distributional critic that learns the full state-action value distribution and therefore takes into account the uncertainty in the transit system as well as the behavior of other agents. For the actor that determines holding control, we train it based on the distorted state-action value distribution from the critic. Particularly, we treat risk preference as a confidence level in our multi-agent setting. For example, the risk-seeking preference of an agent can be interpreted as high confidence in the contribution of its own holding control during the operation horizon. By imposing different confidence levels, we can produce distorted state-action value distributions to reflect agents’ confidence levels in their contributions. By doing so, the model can avoid approximation errors during credit assignment and take into account the uncertainty during the agents’ policy evaluation. In the following, we will introduce the methodology in detail. Note that the proposed framework achieves continuous control based on IQN and meta-learning, we name it as IQNC-M (IQN for Continuous control with Meta-learning). The symbols and notations used to describe the proposed methodology are provided in Table. II.

1) Distributional Critic to Handle Uncertainty: In this study, the critic learns to evaluate the control decision of
each bus. To handle the uncertainty from the transit operation, we use a distributional critic, which models quantile function for state-action value for each state-action pair \((s_t, a_t)\). It should be noted that, in general, independent actor-critic frameworks, the critic only takes local observations for policy evaluation. In order to incorporate global information to promote efficient policy evaluation, we propose a meta-gradient learning scheme in Section IV-B.3. Specifically, state-action value at quantile fraction \(\tau\) is \(Z_\tau (s_t, a_t; \phi)\), where \(\phi\) denotes the parameters of the critic.

Traditionally, the critic learns state-action value expectation by minimizing temporal-difference (TD) error. In view of distributional RL, Dabney et al. [30] proposed adopted quantile functions and proposed implicit quantile network (IQN) to approximate the state-action value distribution. The TD error between two state-action values, on quantile fractions \(\tau_k\) and \(\tau_{k'}\), is:

\[
\delta_{k,k'}^\tau = r_t + \gamma [Z_{\tau_{k'}} (s_{t+1}, a_{t+1}; \phi^\tau^\sim)] - Z_{\tau_k} (s_t, a_t; \phi),
\]

where \(\phi^\tau\) denotes the parameter in the target network [4]. The quantile fractions are randomly sampled from a uniform distribution (i.e., \(\tau_k, \tau_{k'} \sim U (0, 1)\)) and fed into the critic after embedding.

To formulate an efficient objective to train the distributional critic, previous studies suggested using Huber loss due to smooth gradient and proposed to use quantile Huber loss as a surrogate for the Wasserstein distance between two one-dimensional distributions [30]. Specifically, the Huber loss with threshold \(\kappa\) is defined as [41]:

\[
\mathcal{L}_\kappa (x) = \begin{cases} 
\frac{1}{2} x^2 & \text{if } |x| < \kappa, \\
\kappa (|x| - 1 - \frac{x}{2}) & \text{otherwise}, 
\end{cases}
\]

where \(x\) are the residuals. The quantile Huber loss at quantile fraction \(\tau\) is:

\[
\rho_\tau^\kappa (x) = |\tau - \mathbb{I} \{x < 0\}| \frac{1}{\kappa} \mathcal{L}_\kappa (x).
\]

Finally, the objective of the critic using quantile Huber loss function can be approximated by sampling \(K\) and \(K'\) independent quantile fractions:

\[
\mathcal{L} (\phi) = \sum_{k=0}^{K} \sum_{k'=0}^{K'} (\tau_k - \tau_{k'}) \rho_{\tau_k}^\kappa (\delta_{k,k'}). \tag{17}
\]

2) Policy Learning Under Distorted Confidence: Following the traditional actor-critic framework, the actor’s policy is optimized based on the evaluation from the critic. For an ego agent (i.e., the bus whose policy is being evaluated), the critic is proposed to learn the full state-action value distribution (i.e., the quantile value at sampling quantile fractions), then a distorted state-action value expectation is introduced for the actor to improve its policy optimization. The motivation is that: although the distributional critic accounts for the uncertainty in the system dynamic, it only approximates an overall impact after the ego agent’s decision. Thus the critic can not distinguish the contribution of the ego agent and other agents to the system’s current state. To deal with this limitation, we impose distortion weights over the quantiles to represent the confidence in the contribution of the actor on the overall impact. Intuitively, higher weights on larger quantiles indicate stronger confidence in the contribution of the ego agent to the overall impact.

Assuming actor is parameterized as \(\mu (s; \theta)\), it can be trained by maximizing the following objective:

\[
\mathcal{J} (\theta) = \mathbb{E} \left[ \hat{Q} (s_t, \mu (s_t; \theta)) \right], \tag{18}
\]

where the distorted state-action value expectation can be formulated as \(\hat{Q} (s_t, \mu (s_t; \theta)) \approx \sum_{k=0}^{K-1} (\tau_{k+1} - \tau_k) \omega (\hat{\tau}_k) Z_{\tau_k} (s_t, a_t; \phi)\). In Eq. (18), the distortion weights \(\omega (\hat{\tau}_k), k = 0, 2, \ldots, K - 1\) is predefined. In the next subsection we propose a meta-learning scheme on distortion weights to allow a more flexible and robust learning framework.

3) Robust Multi-Agent Control With Meta-Gradient Learning: In this study, meta-learning is introduced to learn adaptive distortion weights to distort state-action value distribution from the critic. To obtain adaptive distortion weights, we utilize the same structure and information similar to the event critic in our previous work [12]. To be specific, the distortion weights are dynamically determined based on the asynchronous events between two consecutive actions of an ego agent (i.e., the holding decisions of other buses). Notably, the state and holding decisions of other buses between two consecutive actions of an ego agent are treated as global information, which
controls to current time step. Thereafter, the distortion weights inform ego bus of what other buses experienced from its last control decision. This scheme is expected to improve policy learning from two aspects. First, while Wang and Sun [12] proposed event graph $G_t$ to incorporate asynchronous control events as global information for accurate credit assignment, they have not fully exploited such information to elevate the robustness of the control policy. To address this limitation, a meta-learning module is introduced to learn adaptive distortion weights taking into account the asynchronous events. Therefore we can promote distributional RL to handle the uncertainty efficiently and therefore obtain better robustness.

Fig. 4. Meta-learning on distortion weights for robust policy learning.

Second, as mentioned in IV-B.2, while the distributional uncertainty is a critical component of the output, the behavior of other agents, the adaptive distortion weights can further express the uncertainty of the transit system and the behavior of other agents, the adaptive distortion weights can further express.

Fig. 5. Structure of IQNC-M for transit control.

We develop and calibrate the bus simulator to reproduce the patterns of real-world operation. In this simulation, the alighting and boarding times per passenger are set to $t_a = 1.8$ sec/pax and $t_b = 3.0$ sec/pax, respectively. To introduce uncertainties of road conditions, buses are given a random speed $v \times \mathcal{U}(0.6, 1.2)$ km/h when traveling between every two consecutive stops, where $v$ is set to 30 km/h and $\mathcal{U}$ denotes a continuous uniform distribution. The capacity of the bus is set to 120 pax. In Fig. 6 (a) and (b), we display the simulated boarding time and actual boarding time from smart card (tap-in for boarding and tap-out for alighting) data and the simulated journey time and actual journey time, respectively, for service R1. The Pearson correlations for these two plots are 0.999 and 0.998.

Table III: Basic Information of Bus Lines

| Services | Stops | Length (km) | Mean (sec) | Std (sec) |
|----------|-------|-------------|------------|-----------|
| R1       | 59    | 46          | 17.4       | 874       | 302       |
| R2       | 72    | 58          | 23.7       | 745       | 307       |
| R3       | 57    | 61          | 23.2       | 931       | 354       |
| R4       | 55    | 46          | 22.5       | 955       | 351       |

V. Experiments

A. Experiments Setup

1) Experimental Data: We evaluate the proposed method based on real-world data. Four bus routes in an Asian city are selected, with actual passenger demand derived from smart card data. The four routes are all trunk services covering more than 15 km, with over 40 bus stops along the route. Table III lists the basic statistics of the routes, including the number of services per day, the number of bus stops on the route, route length, and the mean and standard deviation (std) of headway at the departure terminal.

We develop and calibrate the bus simulator to reproduce the patterns of real-world operation. In this simulation, the alighting and boarding times per passenger are set to $t_a = 1.8$ sec/pax and $t_b = 3.0$ sec/pax, respectively. To introduce uncertainties of road conditions, buses are given a random speed $v \times \mathcal{U}(0.6, 1.2)$ km/h when traveling between every two consecutive stops, where $v$ is set to 30 km/h and $\mathcal{U}$ denotes a continuous uniform distribution. The capacity of the bus is set to 120 pax. In Fig. 6 (a) and (b), we display the simulated boarding time and actual boarding time from smart card (tap-in for boarding and tap-out for alighting) data and the simulated journey time and actual journey time, respectively, for service R1. The Pearson correlations for these two plots are 0.999 and 0.998.
Algorithm 1 IQNC-M for Transit Control

1. Set time horizon $T$ based on real-world operation data.
2. Set memory buffers $\{B_i = \emptyset\}_{i \in N}$, minibatch size as $c$, size to begin training as $B$, number of experience as $C = 0$.
3. Initialize parameters $\phi, \theta, \eta$ for actor, critic and meta learner.
4. Set learning rate $\alpha_a, \alpha_c, \alpha_m$ for actor, critic and meta learner, $K$ the number of quantiles.

for each episode do

1. Set agents’ last activated time $\{t_i^- = 0\}_{i \in N}$
2. Set empty memory stacks $\{\text{Ms}_i, \text{Ma}_i, \text{Mt}_i\}_{i \in N}$
3. for $t = 0$ to $T$ do

   if Buses $b_i \in N$ then
    1. if Bus $b_i$ arrives at bus stop then
     a. Observe $s_{i,t}$ and compute $a_{i,t} = \mu_\theta(s_{i,t})$
     b. $\text{Ms}_i \leftarrow \text{Ms}_i \cup \{s_{i,t}\}$
     c. $\text{Ma}_i \leftarrow \text{Ma}_i \cup \{a_{i,t}\}$
     d. if $t_i^- > 0$ then
        1. $\text{Mt}_i \leftarrow \text{Mt}_i \cup \{t_i^-\}$
       end if
     e. if $t_i^- > 1$ then
        1. $g_{i,t_i^-} = \{(s_{j,t'}, a_{j,t'}, j, i') \in \forall c, j \neq i\}$
       end if
     f. $B_i \leftarrow B_i \cup \{a_{i,t_i^-}, s_{i,t_i^-}, t_i^-, g_{i,t_i^-}\}$
     g. $C = C + 1$
    end if
   end if
   end for

4. if $C > B$ then

   if $B_i \neq \emptyset$ then
    1. Sample buffer $B_i$
    2. Sample $c$ experience from $B_i$
    3. Set $K$ quantiles $\varepsilon_k = \frac{k}{K}, k = 0, \ldots, K$ and $K'$ target quantiles $\varepsilon_k' = \frac{k}{K'}, k' = 0, \ldots, K'$
    4. Generate samples $\text{Z}_{t_i^0}, k = 0, \ldots, K, \text{Z}_{t_i^0}' = 0, \ldots, K'$
    5. Sort the samples $\text{Z}_{t_i^0}, \text{Z}_{t_i^0}'$ in ascending order
    6. Compute $\delta_{k'}$ using Eq. (14)
    7. Update $\phi$ using gradient step $a_c \triangledown \mathcal{L}(\phi)$
    8. Update $\theta$ using gradient step $a_a \triangledown \mathcal{J}(\theta)$
    9. Update $\eta$ using gradient step $a_m \triangledown \mathcal{J}'(\eta)$
   end if
end if

end for

0.983, respectively, suggesting that our simulator is highly consistent with the real-world operation.

2) Models: The proposed IQNC-M framework is compared with the following baseline models, including both traditional headway-based control models and state-of-the-art RL methods. We also consider variants of the proposed model for validation: (i)

1) No control (NC): NC is considered as a naive baseline where no holding control is implemented;
2) Forward headway-based holding control (FH): [20]:

   the holding time is $d = \max \{0, d + g(H_0 - h^-)\}$,

   where $H_0$ is the original headway for dispatch, $h^-$ is forward headway of the arriving bus, $d$ is the average delay at equilibrium, and $g > 0$ is a control parameter. We use the same parameters as in Daganzo [20];
3) Independent Actor-Critic (IAC): [25]: we implement DDPG in the IAC setting to examine the performance where the agent completely overlooks the impact from other agents;
4) Credit Assignment Framework for Asynchronous Control (CAAC): [12]: we implement CAAC as a state-of-the-art MARL method for transit control, which have not specifically designed measures to address robustness issue;
5) IQNC-N: we implement a continuous version of implicit quantile network (IQN) [30], which is a baseline to examine naive distributional RL in a multi-agent setting. The parameters and architectures are the same with IQNC-M apart from the meta-learning module. In addition, there are no distortion weights to indicate the confidence of the policy evaluation.;
6) IQNC-UCF: we implement IQN with distortion risk measure as Eq. (8). This setting indicates that buses are unconfident in their contribution to the transit system along the control horizon. We set $\beta$ in this method is 0.8 to produce the distortion weights. The parameters and architectures are the same as in IQNC-M, however, without the meta-learning module to adjust the distortion weights;
7) IQNC-CF: Similar to IQNC-UCF, we set $\beta = -0.8$ for risk-seeking policy, which can be interpreted as buses are confident in its contribution to the system along the control horizon. The parameters and architectures are the same with IQNC-M but there is no meta-learning module to adjust the distortion weights.

The network architecture of proposed IQNC-M is summarized as follows:

- **Actor:** It is a fully connected network with three hidden layers. Each hidden layer consists of 400 hidden units for each. Output is the holding action. Elu [42] is used for nonlinear activation.
- **Distributional Critic:** The architecture is the same as IQN [30]. Output is the estimation of quantile values. Specifically, we set the hidden layer consists of
400 hidden units. The embedding dimension of quantile fraction $\tau$ is 8 and the number of quantile values $K$ is 16.

- **Meta-Learner:** The architecture is the same as the critic network of CAAC [12]. We modify the last layer for output distortion weights.

The major parameters related to the RL model training is as shown in Table IV.

3) **Evaluation Metrics:** Following [12], we use the following indicators to evaluate model performance:

- **Average waiting time (AWT),** which evaluates the severity of bus bunching;
- **Average travel time (ATT),** which quantifies the average travel time for each bus between the departure terminus to the final terminus;
- **Average occupancy dispersion (AOD),** which evaluates how balanced the occupancy is. Note the dispersion is computed as a variance-to-mean ratio for the occupancy measures. We expect to see a large AOD value when bus bunching happens, as the loading will be highly imbalanced.
- **Average holding duration (AHD),** which characterizes the degree of intervention. We expect less AHD is needed to avoid bus bunching.

4) **Experimental Scenarios:** We conduct robustness analysis by artificially manipulating and disturbing the basic transit demand or traffic state. Specifically, the disturbed transit demand $d_{i,j}$ (pax/hour) with the origin $i$ and the destination $j$ can be modeled as:

$$d_{i,j} = \hat{d}_{i,j} \times p_d,$$

where $\hat{d}_{i,j}$ is the basic demand estimated from smart card data. The scaling factor for demand $p_d$ is sampled from a normal distribution $N_d(1, \sigma_d^2)$ and is clipped within $[1, 10]$. The standard deviation $\sigma_d$ controls the level of uncertainty on the simulated transit demand. Intuitively, by sampling a scaling factor at each simulation for each origin-destination, we obtain an uncertain demand each time. Similarly, we can define the disturbed cruising speed between stops $i$ and $j$ as:

$$s_{i,j} = \hat{s}_{i,j} \times p_s,$$

where $\hat{s}_{i,j}$ (km/s) is the mean cruising speed estimated from smart card data. The scaling factor for speed $p_s$ is sampled from a normal distribution $N_s(1, \sigma_s^2)$ and is clipped within $[0.8, 1.2]$. The standard deviation $\sigma_s$ models the level of uncertainty on the disturbed cruising speed.

To enhance the generalization ability, at each training episode, we impose additional uncertainty by sampling $\sigma_d, \sigma_s$ from a uniform distribution. In this way, we generate different strengths of uncertainties. Notably, this experimental setting indicates that we train and test the model under uncertainty resulted from $\sigma_s$ and $\sigma_d$. We can adjust the uncertainty of $\sigma_s$ and $\sigma_d$ to simulate various uncertain scenarios. Without loss of generality, we conduct experiments with $\sigma_d \sim U[0, 3]$, $\sigma_s \sim U[0, 0.3]$. It should be noted that in different real-world application, more calibration on $\sigma_s$ and $\sigma_d$ is required to construct plausible scenarios for model to learn control policy.

All models are implemented with python and PyTorch 1.7.0 on Ubuntu 18.04 LTS, and experiments are conducted on a server with 256GB RAM. The training cost of the proposed model in the experimental scenario is about one hour.

**B. Result Analysis**

1) **Evaluation on Traffic State Perturbations:** We first fix transit demand uncertainty (i.e., $\sigma_d = 1$) and analyze the performance of control strategies by disturbing the normal traffic state with different levels of perturbations (i.e., $\sigma_s = \{0.1, 0.2, 0.3\}$). We summarize the results of different models in Table V. As can be seen, in general, stronger traffic state perturbations will lead to less reliable and less efficient transit services. Overall, the proposed IQNC-M achieves the best regularity on the system, confirming its robustness under various degrees of traffic state uncertainty. Notably, both the proposed model and CAAC can achieve stable and efficient control performance, but other RL baseline models essentially fail. This is mainly due to the fact that those RL models without utilizing global information suffer from unstable training in the dynamic transit system, especially when additional uncertainty is introduced. This result further confirms the importance of uncertainty in training RL models. Notably, we find that the baseline models based on distributional RL methods (i.e., IQNC-N, IQNC-UCF, IQNC-CF) are also less effective than IQNC-M. We believe this is mainly due to the challenges of training distributional MARL models in the non-stationary multi-agent setting. By introducing the meta-learning module to incorporate global information, the proposed IQNC-M becomes efficient and effective in addressing this issue. We also find that IQNC-UCF implements the least holding control interventions (with minimum AHD). As IQNC-UCF distorts the action-value distribution to place less confidence on the contribution of an ego bus, it turns out that the learned policy will not encourage a long holding period. We suggest that its policy will be less efficient in addressing robustness issue in transit control where headway consistency is the primary consideration. Moreover, from the result on R2-R4, we evaluate the RL models on the route in which no training is conducted. Thus the transferability of the RL models can be validated. Though both CACC and the proposed IQNC-M have utilized global information, IQNC-M outperforms CACC in different bus routes operation. We suggest that it is because the meta-learning module produces adaptive distortion weights. Such a scheme directly results in a better form of state-action value distribution for policy optimization. Although CACC conducts credit assignment to distinguish the contributions of an ego bus from others, this scheme may introduce additional approximation errors and hinder policy learning.

2) **Evaluation on Transit Demand Perturbations:** In this subsection, we fix traffic state uncertainty (i.e., $\sigma_s = 0.1$) and analyze how control strategies perform under different transit demand perturbations (i.e., $\sigma_d = \{1, 2, 3\}$). We conduct comparative analysis similar to traffic state perturbations. As shown in Table VI, it can also be observed that the stronger perturbations in transit demand place a more challenging
scenario to control. The proposed model again performs the best in stabilizing the system (i.e., with smaller AWT and AOD) on both trained and untrained bus routes. It should be pointed out that, similar to the previous result, IQNC-UCF determines less holding period, and the bus will experience less average travel time. However, such a strategy does not perfectly match the original goal of system stability. Besides, it will be less efficient given more severe traffic situations (i.e., anomalous events). In future research, we can design a more comprehensive reward function to balance the trade-off between dwell time serving passengers and the additional holding period.

### TABLE IV

| Hyperparameter      | IQNC-M (proposed) | IQNC-N | IQNC-UCF | IQNC-CF |
|---------------------|-------------------|--------|----------|---------|
| Discount(\(\gamma\)) | 0.9               | 0.9    | 0.9      | 0.9     |
| Minibatch size      | 16                | 16     | 16       | 16      |
| Optimizer           | Adam [43]         | Adam   | Adam     | Adam    |
| Number of quantiles | 16                | 16     | 16       | 16      |
| Initializer         | random uniform    | random uniform | random uniform | random uniform |
| Learning rate (actor, critic, meta learner) | (0.0001, 0.001, 0.0001) | (0.0001, 0.001, /) | (0.0001, 0.001, /) | (0.0001, 0.001, /) |

### TABLE V

**Execution Performance Under Different Strength of Traffic State Perturbations. Model Performance Is Evaluated Using:**
1) **Average Holding Time,** and **Changes In—**
2) **Average Waiting Time** (\(\Delta\text{AWT}\)),
3) **Average Travel Time** (\(\Delta\text{ATT}\)), and
4) **Average Occupancy Dispersion** (\(\Delta\text{AOD}\))—w.r.t. Baseline NC. The Best Results Are Highlighted in Bold.

| Method      | Execution performance with respect to \(\sigma_{s} = \{0.1, 0.2, 0.3\}\) on trained route (R1) |                 |                 |                 |
|-------------|--------------------------------------------------------------------------------------------------|------------------|------------------|------------------|
| AHD (sec)   | \(\Delta\text{AWT} \text{ (sec)}\)                                                                 | \(\Delta\text{ATT} \text{ (sec)}\) | \(\Delta\text{AOD}\) |
| NC          | \(- / - / -\)                                                                                   | 527 / 548 / 578 | 1019 / 1058 / 1147 | 10.5 / 11.3 / 11.7 |
| FH          | 57 / 58 / 61                                                                                     | 1 / -15 / -25   | +575 / +576 / +606 | 0.3 / -0.2 / 0.1  |
| CAAC        | 33 / 32 / 32                                                                                     | -31 / -48 / -56 | +284 / +271 / +255 | -2.6 / -3.4 / -3.3 |
| IAC         | 32 / 31 / 31                                                                                     | -27 / -43 / -31 | +266 / +253 / +236 | -2.3 / -2.9 / -2.7 |
| **IQNC-UCF**| **22 / 21 / 21**                                                                                   | **-26 / -44 / -34** | **+157 / +149 / +145** | **-2.1 / -2.9 / -2.7** |
| IQNC-CF     | 22 / 23 / 23                                                                                     | -33 / -48 / -53 | +155 / +146 / +139 | -2.4 / -3.0 / -2.7 |
| IQNC-M      | 37 / 37 / 39                                                                                     | -46 / -65 / -75 | +319 / +310 / +320 | -3.8 / -4.7 / -4.7 |

| Method      | Execution performance with respect to \(\sigma_{s} = \{0.1, 0.2, 0.3\}\) on untrained route (R2) |                 |                 |                 |
|-------------|--------------------------------------------------------------------------------------------------|------------------|------------------|------------------|
| AHD (sec)   | \(\Delta\text{AWT} \text{ (sec)}\)                                                                 | \(\Delta\text{ATT} \text{ (sec)}\) | \(\Delta\text{AOD}\) |
| NC          | \(- / - / -\)                                                                                   | 553 / 562 / 581 | 1079 / 1113 / 1196 | 17.5 / 17.8 / 18.4 |
| FH          | 62 / 63 / 65                                                                                     | -59 / -67 / -67 | +525 / +523 / +541 | -3.2 / -3.5 / -3.8 |
| CAAC        | 32 / 31 / 31                                                                                     | -105 / -112 / -103 | +186 / +180 / +176 | -7.0 / -7.3 / -6.5 |
| IAC         | 31 / 31 / 30                                                                                     | -94 / -100 / -83 | +180 / +173 / +173 | -6.0 / -6.3 / -5.3 |
| **IQNC-UCF**| **22 / 22 / 21**                                                                                   | **-77 / -83 / -66** | **114 / +110 / +123** | **-5.0 / -5.2 / -4.0** |
| IQNC-CF     | 23 / 23 / 23                                                                                     | -87 / -91 / -77 | +105 / +101 / +106 | -5.7 / -5.8 / -5.0 |
| IQNC-M      | 40 / 40 / 40                                                                                     | -127 / -130 / -129 | +271 / +267 / +272 | -8.5 / -8.7 / -8.5 |

| Method      | Execution performance with respect to \(\sigma_{s} = \{0.1, 0.2, 0.3\}\) on untrained route (R3) |                 |                 |                 |
|-------------|--------------------------------------------------------------------------------------------------|------------------|------------------|------------------|
| AHD (sec)   | \(\Delta\text{AWT} \text{ (sec)}\)                                                                 | \(\Delta\text{ATT} \text{ (sec)}\) | \(\Delta\text{AOD}\) |
| NC          | \(- / - / -\)                                                                                   | 571 / 588 / 632 | 1051 / 1085 / 1169 | 9.9 / 10.3 / 11.6 |
| FH          | 61 / 62 / 64                                                                                     | +22 / +14 / +9   | +753 / +756 / +771 | +0.3 / +0.1 / +0.8 |
| CAAC        | 34 / 34 / 33                                                                                     | -46 / -57 / -69 | +314 / +305 / +294 | -2.1 / -2.3 / -2.8 |
| IAC         | 33 / 33 / 32                                                                                     | -38 / -50 / -63 | +295 / +287 / +279 | -1.7 / -2.0 / -2.5 |
| **IQNC-UCF**| **22 / 23 / 22**                                                                                   | **-34 / -48 / -53** | **+186 / +179 / +180** | **-1.5 / -1.9 / -2.1** |
| IQNC-CF     | 23 / 24 / 24                                                                                     | -41 / -52 / -60 | +195 / +189 / +179 | -1.8 / -1.9 / -2.3 |
| IQNC-M      | 40 / 39 / 40                                                                                     | -55 / -68 / -94 | +411 / +402 / +397 | -2.6 / -3.0 / -3.8 |

| Method      | Execution performance with respect to \(\sigma_{s} = \{0.1, 0.2, 0.3\}\) on untrained route (R4) |                 |                 |                 |
|-------------|--------------------------------------------------------------------------------------------------|------------------|------------------|------------------|
| AHD (sec)   | \(\Delta\text{AWT} \text{ (sec)}\)                                                                 | \(\Delta\text{ATT} \text{ (sec)}\) | \(\Delta\text{AOD}\) |
| NC          | \(- / - / -\)                                                                                   | 580 / 600 / 673 | 788 / 815 / 884 | 6.7 / 7.1 / 8.5 |
| FH          | 60 / 61 / 64                                                                                     | -21 / -31 / -59 | +379 / +380 / +388 | -0.5 / -0.7 / -1.5 |
| CAAC        | 34 / 34 / 33                                                                                     | -68 / -79 / -103 | +150 / +144 / +134 | -2.2 / -2.4 / -2.9 |
| IAC         | 33 / 33 / 32                                                                                     | -62 / -74 / -78 | +143 / +133 / +137 | -1.9 / -2.1 / -2.0 |
| **IQNC-UCF**| **22 / 22 / 22**                                                                                   | **-56 / -65 / -86** | **+88 / +82 / +77** | **-1.7 / -1.7 / -2.0** |
| IQNC-CF     | 23 / 23 / 23                                                                                     | -62 / -70 / -72 | +84 / +82 / +83  | -1.9 / -1.9 / -2.1 |
| IQNC-M      | 39 / 40 / 41                                                                                     | -74 / -90 / -128 | +200 / +197 / +194 | -2.6 / -2.9 / -3.5 |
3) Evaluation on Anomaly: The perturbations may be regarded as a kind of general noise in the transit system, and they are insufficient to represent the uncertainty during real-world transit operations. Notably, the anomaly is another tricky issue in daily transit operations resulting from some special events. While most studies overlook such scenarios, we particularly examine the robustness of control strategies under two of the most common anomaly scenarios (i.e., demand surges and traffic interruptions). Specifically, on top of general perturbation (i.e., $\sigma_d = 1, \sigma_s = 0.1$), we randomly select different number of stops encountering abnormal situations to evaluate the performance. For demand surge on each stop, there will be additional 50 passengers on each bus arrival for a one-hour span. By doing this we simulate the demand surge due to some special events in the neighboring area of these stops (i.e., music concert); for traffic interruptions, the bus speed will be reduced by 90% for a one-hour span after the bus passing the selected stops. Such scenarios may be caused by a downstream traffic accident. We highlight the performance of RL-based methods by comparing their evaluation metric relative to FH. As shown in Fig. 7 and Fig. 8, the most significant difference lies in the holding period, where the RL-based methods tend to adopt less holding period even in severely abnormal situations. Moreover, the proposed method presents a better performance by reducing most passenger waiting costs (i.e., an improvement on AWT) and achieving a more balance bus usage (i.e., an improvement on AOD). Such comparison demonstrates that the proposed method can achieve more robust control under demand surges or traffic interruptions.

| Method | Execution performance with respect to $\sigma_d = \{1, 2, 3\}$ on trained route (R1) | AHD (sec) | AWT (sec) | ATT (sec) | AOD |
|--------|------------------------------------------------------------------|-----------|-----------|-----------|-----|
| NC     | - / - / -                                                          | 527 / 560 / 564 | 1019 / 1071 / 1113 | 10.5 / 13.9 / 15.7 |
| FH     | 57 / 58 / 58                                                       | +1 / -21 / -17 | +575 / +551 / +566 | +0.3 / -1.5 / -0.3 |
| CAAC   | 33 / 32 / 32                                                       | -31 / -60 / -59 | +264 / +263 / +230 | -2.6 / -4.0 / -4.5 |
| IAC    | 32 / 31 / 31                                                       | -27 / -56 / -53 | +266 / +248 / +216 | -2.3 / 3.6 / 3.9 |
| IQNC-N | 22 / 23 / 23                                                       | -27 / -56 / -53 | +159 / +144 / +121 | -2.2 / -3.5 / -3.7 |
| IQNC-UCF | 22 / 22 / 22                                                      | -26 / -52 / -48 | +157 / +144 / +118 | -2.1 / -3.3 / -3.6 |
| IQNC-CF | 23 / 23 / 23                                                       | -33 / -61 / -59 | +155 / +145 / +122 | -2.4 / -3.7 / -4.0 |
| IQNC-M | 37 / 37 / 37                                                       | -46 / -76 / -80 | +319 / +297 / +260 | -5.8 / -5.5 / -6.0 |

| Method | Execution performance with respect to $\sigma_d = \{1, 2, 3\}$ on untrained route (R2) | AHD (sec) | AWT (sec) | ATT (sec) | AOD |
|--------|------------------------------------------------------------------|-----------|-----------|-----------|-----|
| NC     | - / - / -                                                          | 553 / 578 / 656 | 1079 / 1117 / 1151 | 17.5 / 18.9 / 19.3 |
| FH     | 62 / 62 / 62                                                       | -59 / -55 / -52 | +523 / +494 / +484 | -3.2 / -3.3 / -2.4 |
| CAAC   | 32 / 31 / 32                                                       | -105 / -102 / -87 | +136 / +169 / +169 | -7.0 / -7.0 / -5.8 |
| IAC    | 31 / 31 / 31                                                       | -94 / -82 / -69 | +180 / +168 / +170 | -6.0 / -5.8 / -4.6 |
| IQNC-N | 24 / 25 / 26                                                       | -77 / -66 / -47 | +115 / +116 / +137 | -5.0 / -4.6 / -3.4 |
| IQNC-UCF | 22 / 22 / 22                                                      | -77 / -66 / -57 | +104 / +95 / +104 | -5.0 / -4.8 / -3.9 |
| IQNC-CF | 23 / 24 / 25                                                       | -87 / -71 / -57 | +105 / +106 / +126 | -5.7 / -5.3 / -4.2 |
| IQNC-M | 40 / 40 / 40                                                       | -127 / -121 / -103 | +271 / +252 / +260 | -8.5 / -8.3 / -6.8 |

| Method | Execution performance with respect to $\sigma_d = \{1, 2, 3\}$ on untrained route (R3) | AHD (sec) | AWT (sec) | ATT (sec) | AOD |
|--------|------------------------------------------------------------------|-----------|-----------|-----------|-----|
| NC     | - / - / -                                                          | 571 / 653 / 816 | 1051 / 1082 / 1115 | 9.9 / 11.0 / 11.3 |
| FH     | 61 / 61 / 61                                                       | +22 / +25 / +62 | +753 / +732 / +713 | +0.3 / +0.3 / +0.7 |
| CAAC   | 34 / 34 / 34                                                       | -46 / -55 / -30 | +314 / +295 / +291 | -2.1 / -2.2 / -1.6 |
| IAC    | 33 / 33 / 33                                                       | -38 / -45 / -20 | +295 / +282 / +282 | -1.7 / -1.9 / -1.3 |
| IQNC-N | 24 / 24 / 26                                                       | -37 / -41 / -29 | +391 / +195 / +216 | -1.6 / -1.8 / -1.2 |
| IQNC-UCF | 23 / 23 / 23                                                      | -34 / -43 / -31 | +186 / +177 / +184 | -1.5 / -1.8 / -1.1 |
| IQNC-CF | 24 / 24 / 25                                                       | -41 / -49 / -33 | +195 / +198 / +219 | -1.8 / -2.0 / -1.3 |
| IQNC-M | 40 / 39 / 39                                                       | -55 / -63 / -56 | +411 / +389 / +375 | -2.6 / -2.8 / -1.9 |

| Method | Execution performance with respect to $\sigma_d = \{1, 2, 3\}$ on untrained route (R4) | AHD (sec) | AWT (sec) | ATT (sec) | AOD |
|--------|------------------------------------------------------------------|-----------|-----------|-----------|-----|
| NC     | - / - / -                                                          | 580 / 609 / 738 | 788 / 820 / 853 | 6.7 / 7.8 / 9.3 |
| FH     | 60 / 60 / 61                                                       | -21 / -8 / -20 | +739 / +369 / +351 | -0.5 / -0.6 / -0.9 |
| CAAC   | 34 / 34 / 33                                                       | -68 / -72 / -92 | +150 / +137 / +128 | -2.2 / -2.2 / -2.4 |
| IAC    | 33 / 33 / 33                                                       | -62 / -66 / -86 | +143 / +131 / +123 | -1.9 / -1.9 / -2.1 |
| IQNC-N | 25 / 25 / 27                                                       | -56 / -58 / -70 | +88 / +83 / +87 | -1.7 / -1.6 / -1.7 |
| IQNC-UCF | 23 / 22 / 22                                                      | -59 / -53 / -77 | +78 / +74 / +70 | -1.8 / -1.6 / -1.6 |
| IQNC-CF | 23 / 24 / 25                                                       | -62 / -67 / -87 | +84 / +79 / +86 | -1.9 / -1.8 / -1.9 |
| IQNC-M | 39 / 40 / 40                                                       | -74 / -79 / -105 | +200 / +187 / +176 | -2.6 / -2.7 / -3.0 |

TABLE VI
EXECUTION PERFORMANCE UNDER DIFFERENT STRENGTH OF TRANSIT DEMAND PERTURBATIONS. MODEL PERFORMANCE IS EVALUATED USING: 1) AVERAGE HOLDING TIME, AND CHANGES IN—2) AVERAGE WAITING TIME ($\Delta$AWT), 3) AVERAGE TRAVEL TIME ($\Delta$ATT), AND 4) AVERAGE OCCUPANCY DISPERSION ($\Delta$OAD)—W.R.T. BASELINE NC. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.
different control policies based on the same random seed in traffic interruption and demand surge scenarios, respectively. In Fig. 9, four stops are randomly selected to suffer downstream traffic interruption. As can be seen, these abnormal situations tend to ruin system stability if there is no intervention (i.e., NC). Notably, the proposed IQNC-M model presents better control interventions to recover and maintain system performance and stability. In Fig. 10, we randomly simulate demand surges in four stops. From the trajectories, we can observe that the proposed IQNC-M achieves a faster recovery of the system regularity after a demand surge and maintain efficient control along the following horizon.

4) Analysis on Meta Distortion Weights: One of the key contributions of this study is that we designed a meta-learning scheme to generate adaptive distortion weights based on global information. In this way, we achieve efficient distributional RL in a multi-agent transit control setting. While the previous results have demonstrated the efficiency of the proposed meta-learning module, this subsection will further illustrate its
interpretability. As shown in Fig. 11 and Fig. 12, the distortion weights obtained from multiple random seeds on different quantiles are visualized. Specifically, each line indicates a set of distortion weights under a specific number of events (i.e., one indicates only the bus implements holding action during its two consecutive stop arrival). At the beginning of the training (e.g., Fig. 11), the distortion weights corresponding to a different number of events do not present meaningful information since they are nearly the same. In contrast, after 300-episode training (e.g., Fig. 12), we notice that the distortion weights corresponding to a single event increase sharply for larger quantiles. Meanwhile, there is less increase in distortion weights corresponding to more events. This phenomenon demonstrates that the meta-learning module places higher confidence on the contribution of the bus when there are fewer other control events. It makes sense because if there are no other control events between two consecutive holding decisions of the bus, its holding should have a more dominant impact on the dynamic of the system. Therefore the distortion weights should give higher credit to its action.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a distributional MARL framework to achieve robust bus holding control in the daily operation of a bus fleet/route. The proposed IQNC-M framework exploits the distributional RL setting to handle the uncertainties in the transit system. We introduce an interpretable meta-learning module to achieve efficient distributional RL in a non-stationary multi-agent system. Specifically, we introduce distortion weights to distort the state-action value distribution of the critic and derive the weights based on the meta-learning module. By doing so, the policy evaluation becomes more robust and efficient for policy training. Finally, we design a training procedure for transit operation for robust control policy learning. Our experiments on real-world bus services demonstrate that the proposed framework can achieve better and more stable control under various perturbations and anomalies.

There are several directions for future research. First, adversarial reinforcement learning could be one of the promising schemes to enhance the policy robustness [44]. For example, we could model the transit system as an adversary, which is trained to present rational and systematic perturbations/interruptions for the bus agents to develop a more robust control policy. Second, we admit that there is a room to improve the model for addressing robustness issue with at least cost as possible. In this track we are considering enabling overtaking during transit operation or conduct reward engineering to further exploit MARL model. In addition, this study utilizes multiple source of data (i.e., smart car, automatic vehicle location system), which is likely to suffer from data deficiency or anomaly in real-world application. It could be a critical factor to robustness issue. However, how to address such data limitation is out of scope of this study, we expect better transit data assimilation methods can help address this issue. Finally, we believe that this study can also contribute to other applications beyond transit operation. The idea of learning adaptive distortion weights for distribution MARL offers a promising solution to achieve efficient learning while sidestepping the credit-assignment challenge for a general multi-agent system.

REFERENCES

[1] X. J. Eberlein, N. H. M. Wilson, and D. Bernstein, “The holding problem with real-time information available,” Transp. Sci., vol. 35, no. 1, pp. 1–18, Feb. 2001.
[2] Y. Xuan, J. Argote, and C. F. Daganzo, “Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis,” Transp. Res. B, Methodol., vol. 45, no. 10, pp. 1831–1845, Dec. 2011.
[3] K. Gkiotsalitis and E. C. van Berkum, “An analytic solution for real-time bus holding subject to vehicle capacity limits,” Transp. Res. C, Emerg. Technol., vol. 121, Dec. 2020, Art. no. 102815.
[4] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[5] D. Silver et al., “Mastering the game of go without human knowledge,” Nature, vol. 550, no. 7676, pp. 354–359, 2017.
[6] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” IEEE/CAM J. Autom. Sinica, vol. 3, no. 3, pp. 247–254, Apr. 2016.
[7] T. Cha, J. Wang, L. Codecà, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 1086–1095, Mar. 2019.
[8] F. Bellitti, D. Haziza, G. Gomes, and A. M. Bayen, “Expert level control of ramp metering based on multi-task deep reinforcement learning,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 4, pp. 1198–1207, 2017.
[9] W. Chen, K. Zhou, and C. Chen, “Real-time bus holding control on a transit corridor based on multi-agent reinforcement learning,” in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2016, pp. 100–106.
[10] F. Alesiani and K. Gkiotsalitis, “Reinforcement learning-based bus holding for high-frequency services,” in Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2018, pp. 3162–3168.
[11] J. Wang and L. Sun, “Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework,” Transp. Res. C, Emerg. Technol., vol. 116, Jul. 2020, Art. no. 102661.
[12] J. Wang and L. Sun, “Reducing bus bunching with asynchronous multi-agent reinforcement learning,” in Proc. Int. Joint Conf. Artif. Intell., 2021, pp. 426–433.

[13] M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung, “Learning implicit credit assignment for cooperative multi-agent reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 11853–11864.

[14] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, “Meta-reinforcement learning of structured exploration strategies,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 5307–5316.

[15] O. Cats, A. N. Larijani, H. N. Koutroupolou, and W. Burghout, “Impacts of holding control strategies on transit performance: Bus simulation model analysis,” Transp. Res. Rec., vol. 2216, no. 1, pp. 51–58, 2011.

[16] W. Wu, R. Liu, and W. Jin, “Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour,” Transp. Res. B, Methodol., vol. 104, pp. 175–197, Oct. 2017.

[17] M. D. Hickman, “An analytic stochastic model for the transit vehicle holding problem,” Transp. Sci., vol. 35, no. 3, pp. 215–237, 2001.

[18] F. Delgado, J. Muñoz, R. Giesen, and A. Cipriano, “Real-time control of buses in a transit corridor based on vehicle holding and boarding limits,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2090, no. 1, pp. 59–67, 2000.

[19] S. J. Berrebi, K. E. Watkins, and J. A. Laval, “A real-time bus dispatching policy to minimize passenger wait on a high frequency route,” Transp. Res. B, Methodol., vol. 81, pp. 377–389, Nov. 2015.

[20] C. F. Daganzo, “A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons,” Transp. Res. B, Methodol., vol. 43, no. 10, pp. 913–921, 2009.

[21] S. J. Berrebi, E. Hans, N. Chiabaut, J. A. Laval, L. Leclercq, and K. E. Watkins, “Comparing bus holding methods with and without real-time predictions,” Transp. Res. C, Emerg. Technol., vol. 87, pp. 197–211, Feb. 2018.

[22] E. van der Werff, N. van Oort, O. Cats, and S. Hoogendoorn, “Robust control for regulating frequent bus service: Supporting the implementation of headway-based holding strategies,” Transp. Res. Rec., vol. 2673, no. 9, pp. 654–665, 2019.

[23] S. Li, R. Liu, L. Yang, and Z. Gao, “Robust dynamic bus controls considering delay disturbances and passenger demand uncertainty,” Transp. Res. B, Methodol., vol. 123, pp. 88–109, May 2019.

[24] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[25] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–14.

[26] R. Bellman, “A Markov decision process,” J. Math. Mech., vol. 6, no. 5, pp. 679–684, 1957.

[27] R. S. Sutton et al., “Policy gradient methods for reinforcement learning with function approximation,” in Proc. Adv. Neural Inf. Process. Syst., 1999, pp. 1057–1063.

[28] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 387–395.

[29] M. G. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 449–458.

[30] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, “Implicit quantile networks for distributional reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 1096–1105.

[31] N. A. Urpi, S. Curi, and A. Krause, “Risk-averse offline reinforcement learning,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–17.

[32] J. Dhaene, A. Kukush, D. Linders, and Q. Tang, “Remarks on quantiles and distortion risk measures,” Eur. Actuarial J., vol. 2, no. 2, pp. 319–328, 2012.

[33] S. S. Wang, “A class of distortion operators for pricing financial and insurance risks,” J. Risk Insurance, vol. 67, no. 1, pp. 15–36, 2000.

[34] Z. Xu, H. van Hasselt, and D. Silver, “Meta-gradient reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 2402–2413.

[35] A. Gupta, B. Eysenbach, C. Finn, and S. Levine, “Unsupervised meta-learning for reinforcement learning,” CoRR, vol. abs/1806.04640, pp. 1–12, Jun. 2018.

[36] M. L. Liman, “Markov games as a framework for multi-agent reinforcement learning,” in Machine Learning Proceedings 1994. Amsterdam, The Netherlands: Elsevier, 1994, pp. 157–163.

[37] T. R. Board, Transit Capacity and Quality of Service Manual, 3rd ed. Washington, DC, USA: The National Academies Press, 2013. [Online]. Available: https://www.nap.edu/catalog/24766/transit-capacity-and-quality-of-service-manual-third-edition

[38] J. Yiu, D. Chen, W. Zhao, and L. Chen, “Online adjusting subway timetable by Q-learning to save energy consumption in uncertain passenger demand,” in Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC), Oct. 2014, pp. 2743–2748.

[39] H. Khadilkar, “A scalable reinforcement learning algorithm for scheduling railway lines,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 2, pp. 727–736, Feb. 2019.

[40] N. A. Urpi, S. Curi, and A. Krause, “Risk-averse offline reinforcement learning,” CoRR, vol. abs/2102.05371, pp. 1–17, Feb. 2021.

[41] P. J. Huber, “Robust estimation of a location parameter,” Ann. Math. Statist., vol. 35, pp. 73–101, 1964.

[42] D. -A. Clevert, T. Unterhiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2015, arXiv:1511.07289.

[43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[44] X. Pan, D. Seita, Y. Gao, and J. Canny, “Risk averse robust adversarial reinforcement learning,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 8522–8528.

Jiawei Wang received the B.S. degree in traffic engineering and the M.S. degree in traffic information and control from Sun Yat-sen University, Guangzhou, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Civil Engineering, McGill University, under the supervision of Prof. Lijun Sun. His current research interests include intelligent transportation systems, traffic control, and machine learning.

Lijun Sun (Senior Member, IEEE) received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree in civil engineering (transportation) from the National University of Singapore, in 2015. He is currently an Assistant Professor with the Department of Civil Engineering, McGill University, Montreal, QC, Canada. His research interests include intelligent transportation systems, machine learning, spatiotemporal modeling, travel behavior, and agent-based simulation.