IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 March 2023, accepted 18 March 2023, date of publication 27 March 2023, date of current version 31 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261900

RESEARCH ARTICLE

Intelligent Energy-Efficient Train Trajectory Optimization Approach Based on Supervised Reinforcement Learning for Urban Rail Transits

GUANNAN LI[1,2], **(Student Member, IEEE), SIU WING OR**[1,2], **AND KA WING CHAN**[1], **(Member, IEEE)**

[1]Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[2]Hong Kong Branch of National Rail Transit Electrification and Automation Engineering Technology Research Center, Hong Kong, China

Corresponding author: Siu Wing Or (eeswor@polyu.edu.hk)

This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (HKSAR) Government under Grant R5020-18, and in part by the Innovation and Technology Commission of the HKSAR Government to the Hong Kong Branch of National Rail Transit Electrification and Automation Engineering Technology Research Center under Grant K-BBY1.

ABSTRACT Artificial intelligence of things (AIoT)-enabled intelligent automatic train operation (iATO) is an urgently needed technology to expand the capability of ATO in addressing the real-time responsiveness and dynamic online challenges to energy-efficient train trajectory optimization (TTO) and its associated ride-comfort, punctuality, and safety issues in modern urban rail transit networks. This paper proposes a three-step supervised reinforcement learning-based intelligent energy-efficient train trajectory optimization (SRL-IETTO) approach for iATO by hybrid-integrating deep reinforcement learning (DRL) and supervised learning. First, multiple objectives are formulated based on real-time train operation and systematically integrated into the RL algorithm by a binary function-based goal-directed reward design method. Second, an IETTO model is established to handle uncertain disturbances in real-time train operation and generate optimal energy-efficient train trajectories online by optimizing energy efficiency and receiving supervisory information from trajectories of pre-trained TTO models. Finally, numerical simulations are implemented to validate the effectiveness of the SRL-IETTO using in-service subway line data. The results demonstrate the superiority and improved energy saving of the proposed approach and confirm its adaptability to online trip time adjustments within the practical running time range under uncertain disturbances with less trip time error compared to other intelligent TTO algorithms.

INDEX TERMS Deep reinforcement learning, energy-efficient train trajectory optimization, intelligent automatic train operation, supervised reinforcement learning, urban rail transits.

I. INTRODUCTION

A. MOTIVATION

Urban rail transit networks (URTNs), such as metros, light rails, etc., play an essential role in public transportation in high-density urban areas because of their practical viability in providing sustainable, green, effective, and convenient mass transportation services [1]. The growing needs and challenges to address passenger demand, urban traffic congestion, environmental cleanliness, and sustainability have urged the expansion of the URTNs worldwide [2]. The global urban rail passenger traffic will reach a historical high of 954 billion passenger-kilometers by 2025 [3]. Therefore, the current and near-future URTNs must handle the continuously increasing busy lines and shorter train departure headways while achieving energy-efficient train operation.

From a practical perspective, real-time train operation is frequently affected by disturbances before departure and during operation [4], [5]. These include, but are not limited to, departure delays, arrival delays, and critical demands during

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

peak hours caused by weather, passenger, and equipment. Considering the increasingly short trip times and frequent disturbances in high-density urban areas, even short delays that last for several seconds may result in a secondary delay after the initial delay of the train and even knock-on delays [6]. Moreover, disturbances may lead to extra energy usage and decreased ride comfort due to the temporary train acceleration or deceleration to guarantee punctuality, adversely affecting energy consumption and passenger satisfaction [6]. Importantly, manual driving is difficult to find an optimal train trajectory that satisfies the correspondingly increasing operational requirements.

In the past decades, automatic train operation (ATO) has been proposed and implemented to evolve manual driving into automated driving [7] by automatically determining the optimal train trajectories between stations through train trajectory optimization (TTO). However, the optimal train trajectories of current ATOs are usually designed and optimized in advance [8], [9]. The insufficient capabilities of the ATO-based technologies in terms of calculation and adjustment of energy-efficient trajectories online in response to uncertain disturbances and adapting to rescheduled trip times from the real-time timetable rescheduling (RTTR) [10] significantly limit their application prospects. Therefore, it is necessary to address the energy-efficient TTO and its associated ride-comfort, punctuality, and safety issues under uncertain disturbances and rescheduled trip times to enhance ATO performances.

Thanks to the recent groundbreaking in key enabling and synergetic technologies, including smart sensors, low-power embedded systems, long-range lower-power wireless networks, artificial intelligence (AI), and big data analytics, the artificial intelligence of things (AIoT)-enabled intelligent automatic train operation (iATO) has become practically viable and valuable as the next-generation emerging ATO technology. Specifically, precise real-time train operation information and historical data can be utilized by such an iATO technology to achieve intelligent energy-efficient train trajectory optimization (IETTO) with real-time responsiveness.

B. BACKGROUND

The comprehensive review of the TTO can be found in [8] and [11]. The research of the TTO can be traced back to 1968 by introducing Pontryagin's maximum principle (PMP) [12]. The optimal train trajectory usually contains a sequence of four optimal control regimes and their switching points: maximum acceleration, cruising, coasting, and maximum deceleration. Based on PMP, Howlett [13] found the necessary conditions for an optimal control strategy and developed key equations that determine the optimal switching points. Albrecht et al. [14] used perturbation analysis to deduce the uniqueness of the global optimal strategy and reported the algorithm implementation in *Energymiser*. Alternatively, numerical algorithms [8] can balance the optimization performance and computational time. Haahr et al. [15] applied dynamic programming (DP) to generate optimal speed profiles and relied on an event-based decomposition to reduce the search space. Wang and Goverde [16] used the pseudospectral method to transform the multi-phase optimal control problem into a nonlinear programming (NLP) model and solved the TTO with signal influences.

Some studies [17], [18], [19] have dealt with disturbances in designing optimal train trajectories, usually with punctuality and energy consumption as the two main objectives. Wang et al. [17] developed an approximate dynamic programming method with an online search process for the TTO, considering the uncertainty in traction force and train resistance. Fernández-Rodríguez et al. [18] proposed a robust train trajectory design method considering train load variations and delays. The optimal train trajectories were selected from a Pareto front by particle swarm optimization algorithm. Yang et al. [19] used an evolutionary algorithm to solve a two-phase stochastic programming model with uncertain train mass to optimize train trajectory and timetable. In addition, the TTO has been taken into account in timetable rescheduling [20], [21]. Wang and Goverde [22] developed a multi-train trajectory optimization method on single-track lines to get an adjusted timetable and a set of speed trajectories. The delay and energy consumption were reduced. Dong et al. [23] proposed a parallel intelligent system for train operation control and dynamic scheduling involving a broad information exchange between timetable rescheduling and train operation, which raises high requirements for the TTO.

With the enabling of AI [24], reinforcement learning (RL) has been a powerful methodology for addressing decision-making problems in real-time TTO with disturbances. Yin et al. [25] developed two intelligent train operation algorithms, and the simulations showed that the RL-based algorithm is capable of adjusting trip time dynamically between two stations. They [6] further proposed a Q-learning-based algorithm for the TTO under online adjusted timetables. However, disturbances are typically assumed to be short (within 20 s), and the optimal train trajectories in a broad trip time adjustment range against disturbances have not been fully contemplated. Compared to model-based approaches, the model-free RL-based approaches show significant advantages in self-adaptability and the ability to learn from historical data. Moreover, deep reinforcement learning (DRL) [26] can learn optimal controls in more complicated decision-making problems than conventional RL by using deep neural networks (DNNs), such as settings with high-dimensional state and action spaces. Many studies have combined DRL with expert knowledge rules to solve the TTO [9], [27], [28], [29]. Nevertheless, few consider real-time TTO with disturbances during operation.

Recently, Supervised reinforcement learning (SRL) has been used in robotics and the healthcare domain [30], [31], [32]. Usually, SRL combines supervised learning (SL) with DRL architecture to form a supervisor-actor-critic (SAC)

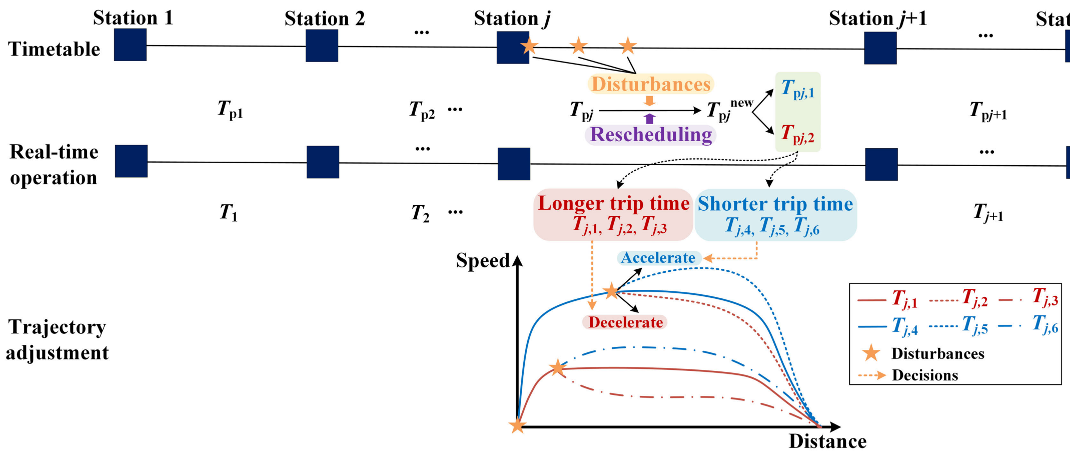


FIGURE 1. Real-time train operation process with disturbances.

architecture. Under the SAC architecture, the supervisor helps optimize the actor's policy by sending low-risk actions as supervisory information to modify the policy update of the actor. The solution strategies and intentions conveyed by the supervisor efficiently accelerate DRL training with fewer learning trials and further improve model performance [33].

C. CONTRIBUTION

In this paper, a supervised reinforcement learning-based intelligent energy-efficient train trajectory optimization (SRL-IETTO) approach is proposed for iATO in the real-time train operation of modern URTNs. The main contributions can be summarized as follows:

1) Multiple objectives associated with energy saving, ride comfort, punctuality, and safety are formulated as trajectory evaluation indices based on real-time train operation. Then they are systematically integrated into the RL algorithm by a goal-directed reward design method based on binary function. A fine-tuning process based on bilinear programming is used to correct RL reward parameters toward their optimal states.

2) An IETTO model using two-step training is developed to handle uncertain disturbances in real-time train operation and generate optimal energy-efficient train trajectories online. The first step generates pre-trained TTO models to obtain fixed-time train trajectories based on the twin-delayed deep deterministic policy gradient (TD3) algorithm. The second step improves model generalization capability within the practical running time range by considering supervision loss between the policy of the IETTO model and pre-trained models into the policy gradient (PG).

The rest of this paper is organized as follows. Section II illustrates the problem formulation with the problem statement followed by the train control model formulation. Section III presents the SRL-IETTO approach, including the SRL principle, the SRL-IETTO framework, and detailed model design and training. Section IV reports the simulation setup and results. Section V gives the conclusions.

II. PROBLEM FORMULATION

A. PROBLEM STATEMENT

Fig. 1 illustrates the IETTO problem. Before the real-time operation, an extensive planning process generates a timetable that regulates the planned trip times $T_{\mathrm{p}\,1}, \ldots, T_{\mathrm{p}j}, \ldots, T_{\mathrm{p}n-1}$ of each trip between stations. According to the predetermined timetable and line data, the train trajectory is optimized in advance. However, real-time operation is frequently affected by disturbances that occur when the train is at the station (before departure) and during operation. Besides, the timetable may be rescheduled. To ensure punctuality, the planned trip time $T_{\mathrm{p}j}$ between station j and $j+1$ can be modified to $T_{\mathrm{p}j}^{\mathrm{new}}$ in real-time operation, which can be a shorter planned arrival time $T_{\mathrm{p}j,1}$ or a longer planned arrival time $T_{\mathrm{p}j,2}$.

Based on the modified planned trip time, the train trajectory should be re-optimized online and meet multiple objectives. In detail, such re-optimized train trajectory should maintain high punctuality subject to uncertain disturbances or timetable rescheduling, high safety such that the on-road speed limit is not violated and stopping accuracy is guaranteed, and certain levels of energy saving and ride comfort for passengers. Based on the re-optimized train trajectory, decisions are made to accelerate or decelerate the train. Correspondingly, the actual running times during the trip can be described as $T_{\,1}, \ldots, T_j, \ldots, T_{n-1}$.

B. TRAIN CONTROL MODEL FORMULATION

Generally, a train can be considered a single-point mass model subject to various forces during operation. Based on Newton's law, the motion of the train can be described as

$$M\rho_{\mathrm{r}}v(x)\frac{\mathrm{d}v(x)}{\mathrm{d}x} = F(v,x) - R_{\mathrm{b}}(v) - R_{\mathrm{l}}(x), \tag{1}$$

$$\frac{\mathrm{d}t(x)}{\mathrm{d}x} = \frac{1}{v(x)}, \tag{2}$$

$$R_{\mathrm{b}}(v) = c_1 + c_2 v + c_3 v^2, \tag{3}$$

$$R_1(x) = Mg\sin(\alpha(x)), \tag{4}$$

where x is the train position, $v(x)$ is the train speed at position x, $t(x)$ is the time at position x, M is the mass of the train, ρ_r is the rotating mass factor, $F(v, x)$ is the traction or braking force, $R_b(v)$ is the basic resistance at speed v, $R_l(x)$ is the line resistance at position x, $\alpha(x)$ is the slope angle at position x. According to the Davis formula, $R_b(v)$ is a quadratic function of speed, where c_1, c_2, and c_3 are the coefficients of train characteristics.

Train operation is subject to several limitations for motor power, speed, and punctuality, which include

$$u_{\text{lim}}^{\text{dcc}} \leq u(x) \leq u_{\text{lim}}^{\text{acc}}, \ \forall x \in [0, x_e], \tag{5}$$

$$v(x) \leq v_{\text{lim}}, \ \forall x \in [0, x_e], \tag{6}$$

$$v(0) = 0, \ v(x_e) = 0, \ t(0) = 0, \ |T_p - T| \leq T_{\text{lim}}, \tag{7}$$

where $u(x)$ is the acceleration at position x, $u_{\text{lim}}^{\text{dcc}}$ and $u_{\text{lim}}^{\text{acc}}$ are the deceleration and acceleration limits, respectively, v_{lim} is the on-road speed limit, x_e is the end position where the train stops, T_{lim} is the punctuality tolerance which regulates the maximum allowed trip time error, T_p is the planned trip time, $T = t(x_e)$ is the actual running time during the trip, $|T_p - T|$ is the absolute difference between the planned trip time and the actual running time, which is the trip time error.

We use $\mathcal{D}^{\text{bd}}$ and $\mathcal{D}^{\text{do}}$ to denote disturbances or rescheduling commands that occur before departure and during operation, respectively. The uncertain trip time changes caused by $\mathcal{D}^{\text{bd}}$ and $\mathcal{D}^{\text{do}}$ are defined as t_d^{bd} and t_d^{do}, respectively. Suppose the disturbances occur or rescheduled trip times are given, and the train receives notifications at position $x_d \in [0, x_e)$. The modified planned trip time is T_p^{new}. Thus,

$$T_p^{\text{new}} = \begin{cases} T_p - t_d^{\text{bd}}, & t_d^{\text{bd}} \sim \mathcal{T}^{\text{bd}}, \text{ if } \mathcal{D}^{\text{bd}} \\ T_p - t_d^{\text{do}}, & t_d^{\text{do}} \sim \mathcal{T}^{\text{do}}, \text{ if } \mathcal{D}^{\text{do}} \end{cases}, \forall x \in [x_d, x_e], \tag{8}$$

$$|T_p^{\text{new}} - T| = |T_p - (T + t_d^{\text{bd}} \text{ or } T + t_d^{\text{do}})| \leq T_{\text{lim}}, \tag{9}$$

where $\mathcal{T}^{\text{bd}}$ and $\mathcal{T}^{\text{do}}$ are the time error distribution of $\mathcal{D}^{\text{bd}}$ and $\mathcal{D}^{\text{do}}$, respectively.

III. SRL-IETTO APPROACH

A. SRL PRINCIPLE

We first introduce the principle of DRL and explain how supervisory information is used to transform DRL into SRL. The principle of DRL can be described as an iterative agent-environment interaction process where the agent learns to make and adjust decisions from the feedback of the environment. For real-time train operation, the agent is the onboard controller of the train, and the environment is the train operation process. The train operation can be treated as a Markov decision process (MDP). At each step i, $0 \leq i \leq N$, the DRL agent takes an action a according to a policy $\mu(s)$ at the current state s in the environment $\mathcal{E}$, and observes a new state s' and a reward r, where N is the maximum step. s denotes the state, a state space $\mathcal{S}$ contains all possible states, $s \in \mathcal{S}$; a denotes the action, an action space $\mathcal{A}$ contains all valid actions, $a \in \mathcal{A}$; r is the immediate reward that evaluates a, a reward space $\mathcal{R}$ contains all rewards, $r \in \mathcal{R}$; $\mu(s) \rightarrow a$ is the probability distribution over a given s.

A sequence of state-action pair (s, a) during multiple steps creates a trajectory ξ (namely, an episode), and it can be evaluated by a return $R(\xi) = \sum_{i=1}^{N} \gamma^{i-1} r_i$, where γ is the discount factor [9]. The goal of the agent is to learn an optimal policy μ^* that generates a sequence of optimal actions that maximizes the expected return. The optimal trajectory ξ^* is found by executing the optimal actions sequentially. The expected return at (s, a) under μ^* is $Q^{\mu^*}(s, a)$, which can be calculated by [34]

$$Q^{\mu^*}(s, a) = \mathbb{E}_{r,s' \sim \mathcal{E}}[r + \gamma \max_{a'} Q^{\mu^*}(s', a')], a' \sim \mu(s'). \tag{10}$$

Due to the complexity of calculating $Q^{\mu^*}(s, a)$ when state and action spaces become large, DRL uses the deep neural network (DNN) as a function approximator to approximate $Q^{\mu^*}(s, a)$ by $Q_\theta(s, a)$ with parameter θ.

In this paper, the SRL uses the supervisory information provided by pre-trained supervisors to give hints on the decisions of the agent. The hints refer to the differences between the policy $\mu(s)$ of the agent and the policy $\mu_{\text{sl}}(s)$ of supervisors. Therefore, by minimizing these differences, $\mu(s)$ is more similar to $\mu_{\text{sl}}(s)$. In other words, the agent learns from interactions with the environment and supervisors.

B. SRL-IETTO FRAMEWORK

The proposed SRL-IETTO approach has three steps: the model design step, the model training step, and the model verification step (see Fig. 2). At the model design step, essential elements of the SRL environment, including state, action, and reward, are designed by extracting features from the train operation process and considering multiple objectives. First, train operation features, including train operation states and constraints, are extracted. Second, multiple objectives associated with energy saving, ride comfort, punctuality, and safety are formulated to establish evaluation indices for optimal train trajectory. Third, the essential elements of the SRL environment are designed. Action a and state s are defined based on operation states. The range of a is adjusted according to the limitations of various constraints to avoid unreasonable actions. For reward r, the evaluation indices are integrated into the reward design to consider multiple objectives.

At the model training step, the SAC architecture is adopted with supervisor pre-training and agent training. First, TTO models are pre-trained by DRL through the standard agent-environment interactions to serve as supervisors. Since disturbances or rescheduled trip times are strongly related to the flexibility of trip time, the agent can learn from multiple optimal train trajectories on different planned trip times to improve its generalization capability. Based on this idea, multiple supervisors, with each of them having a fixed but different planned trip time T_p within the practical running

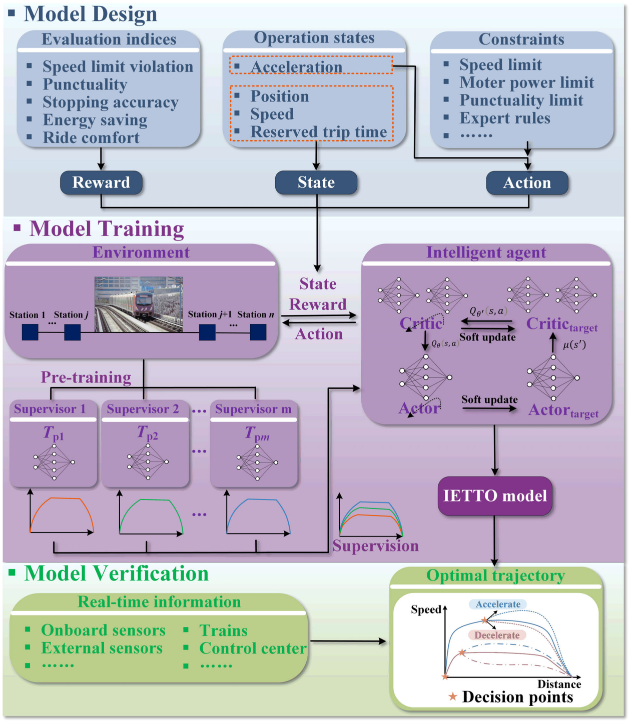


FIGURE 2. SRL-IETTO framework.

time range, are used to provide supervisory information for subsequent agent training. These supervisors are pre-trained without using human driving data or ATO reference trajectories, which avoids the prior data collection. Second, an intelligent agent is trained under the supervision of pre-trained supervisors. Taking advantage of the SRL in training acceleration and model performance improvement and the DNN in dealing with high-dimensional continuous state and action spaces, the intelligent agent can learn to *a*) take the state s as input; *b*) generate continuous and optimal action a as output for accurate train control.

At the model verification step, the well-trained agent, namely the IETTO model, is tested by various cases that simulate real-world situations to verify its model performance and illustrate its practical usage.

C. MODEL DESIGN

1) OPERATION STATES, CONSTRAINTS, AND EVALUATION INDICES

The train operation states involve the train position, speed, reserved trip time [29], and acceleration. Constraints include (5)-(9) and rules derived from expert knowledge of experienced drivers and ATOs. For convenience, we use b to denote constraints. The expert knowledge rules aim to enhance safety, limit the action space, and reduce the complexity of the problem. The rules used are summarized as follows:

a) A safe braking distance $\Delta x = \frac{-v_{\text{lim}}^2}{2u_s}$ is defined, where u_s is the minimum deceleration and is usually used for emergency brakes. We choose $u_s = -1$ m/s^2 [9] for Δx calculation. Once the distance between the current train position x and the next station is less or equal to Δx, the train must decelerate in a constant u.

b) Whenever the speed of the train reaches 95% of the speed limit, the train should not accelerate anymore.

The optimal train trajectory generated by the proposed approach should be evaluated in various aspects. Typical evaluation indices are safety evaluation index (includes on-road speed limit violation evaluation index v_s and stopping accuracy evaluation index Δp), punctuality evaluation index Δt, energy-saving evaluation index E, and ride-comfort evaluation index C [8]. The absolute difference between the actual distance and the position where the train stops indicate the stopping accuracy. For comparison purposes, we use a similar definition in [9] for index C. The indices are

$$v_s = \begin{cases} 1, & \text{if } v \geq v_{\text{lim}} \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

$$\Delta p = |x_e - d|, \tag{12}$$

$$\Delta t = |T_p^{\text{new}} - T|, \tag{13}$$

$$E = \frac{1}{Md}\int_0^{x_e} F(v,x)\mathrm{d}x, \tag{14}$$

$$C = \int_0^{T_p^{\text{new}}} \begin{cases} |\frac{\mathrm{d}u}{\mathrm{d}t}|\mathrm{d}t, & |\frac{\mathrm{d}u}{\mathrm{d}t}| > 0.3 \text{ m/s}^3 \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where d is the actual distance between stations.

2) STATE *s* AND ACTION *a*

The acceleration of the train is defined as action a. For the agent, its state s contains train position, speed, and reserved trip time. For supervisors, their state s^s only contains train position and speed. This is because they do not need to observe T_p since it is fixed. The initial state is defined as $s_0 = [0, 0, T_p]$ and $s_0^s = [0, 0]$.

$$s = [x, v, T_p - T], \tag{16}$$

$$s^s = [x, v]. \tag{17}$$

3) REWARD *r*

Reasonable rewards are designed by adopting a goal-directed reward design method based on binary function [35]. The rewards can be classified into goal-state rewards r_g (namely, for this paper, final-state rewards) and rewards per step r_∞

$$r = \begin{cases} r_g, & s' = s_g \\ r_\infty, & \text{otherwise}, \end{cases} \tag{18}$$

where s_g is the goal state (final state).

Thus, the solution of (10) becomes a constant Q_∞ when $r_g = r_\infty$,

$$Q_\infty = \sum_{i=1}^{N} \gamma^{i-1} r_\infty = r_\infty \frac{1-\gamma^i}{1-\gamma} \approx \frac{r_\infty}{1-\gamma}. \tag{19}$$

If $r_g > Q_\infty$, the final-state rewards are more attractive than rewards in other states, leading the agent to the final state.

TABLE 1. Rewards.

Item	r_∞	r_{g}
–	$+2.5$	$+350$
Speed limit	-1, if $v \geq v_{\mathrm{lim}}$	–
Punctuality	-0.5, if $\Delta t \geq T_{\mathrm{lim}}$	$\begin{cases} r_{\mathrm{T}}\Delta t, & \Delta t \geq T_{\mathrm{lim}} \\ +50, & \text{otherwise} \end{cases}$
Energy	-0.5, if $E \geq E_{\mathrm{lim}}$	$r_{\mathrm{E}}E$
Ride comfort	-0.5, if $C \geq C_{\mathrm{lim}}$	$r_{\mathrm{C}}C/N$
Stopping accuracy	–	$-\Delta p$, if $\Delta p \geq p_{\mathrm{lim}}$

Besides, r_{g} and r_∞ must be non-negative to encourage the agent to move from the current state to the next state. Thus, the relationship between r_{g} and r_∞ is established as

$$r_{\mathrm{g}} > \frac{r_\infty}{1-\gamma} \geq 0. \tag{20}$$

Thus, according to (20), we design different types of rewards following the binary reward function form to reflect various real-world objectives. Table 1 illustrates the designed rewards. For r_∞, the on-road speed limits, punctuality, energy, and ride comfort are considered. These rewards in r_∞ can give immediate feedback at every step to accelerate training. The agent gets penalties once its performance is worse than the tolerance T_{lim}, E_{lim}, and C_{lim} set on punctuality, energy, and ride comfort, respectively. Since safety is the basic operation requirement and most important objective, the penalty weights are higher than other objectives. A bias term is used to ensure the non-negative nature of r_∞. T_{lim} is set to be 3 s [9]. E_{lim} is set to be equal to the practical energy consumption of the same line since we expect better energy saving in agent performance than in practice. C_{lim} is set to be 0.3 g/s [36]. For r_{g}, we design rewards for punctuality, ride comfort, energy, and stopping accuracy. The stopping accuracy tolerance p_{lim} is set to be 0.3 m [37]. The coefficients of various r_{g} terms have significant impact on model performance. Therefore, the coefficient values are fine-tuned toward optimal states (see Appendix).

D. MODEL TRAINING

1) OVERVIEW

The TD3 algorithm [38] enlightens the model training architecture (see Fig. 3). TD3 is an advanced DRL algorithm that uses DNN as the function approximator to handle high-dimensional continuous state and action spaces. It implements techniques such as double Q learning, delayed policy updates, and target policy smoothing to address $Q_\theta(s, a)$ overestimation.

a) Pre-training: each supervisor is trained with a fixed but different T_{p} within the practical running time range. This range is determined by calculating the minimum and maximum planned trip time $T_{\mathrm{p}}^{\min}$ and $T_{\mathrm{p}}^{\max}$ of the trip. The calculation of $T_{\mathrm{p}}^{\min}$ and $T_{\mathrm{p}}^{\max}$ can be referred to [39], where $T_{\mathrm{p}}^{\max}$ is based on the assumption that the lowest average running speed of 40 km/h offered to passengers. $T_{\mathrm{p}1}, T_{\mathrm{p}\,2}, \ldots, T_{\mathrm{p}m}$ for supervisor 1, 2, …, m are uniformly sampled from $T_{\mathrm{p}}^{\min}$ to $T_{\mathrm{p}}^{\max}$. The data $(s, a, s', r, b, b', done)$ of each MDP step are stored and sampled from an experience replay buffer, where *done* is a binary variable which tells whether the terminate state has occurred. TD3 with the prioritized experience replay (PER) buffer [40] is used to train supervisors. PER uses importance sampling to obtain better model training performance.

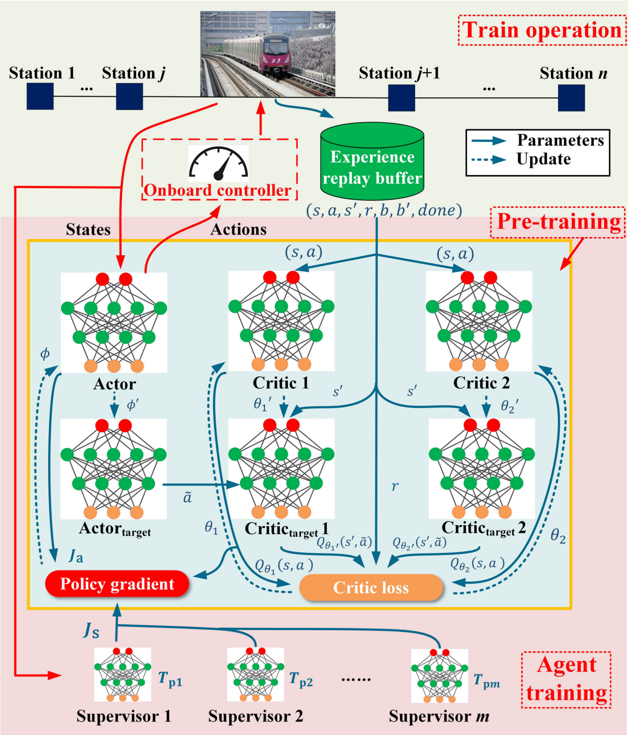


FIGURE 3. Illustration of the proposed model training architecture.

b) Agent Training: the actor outputs action a based on its policy μ, which is updated according to the Q value. The critic estimates Q value. The supervisor calculates the supervision loss, namely, the differences between the supervisor's policy $\mu_{\mathrm{sl}}(s)$ and the agent's policy $\mu(s)$, to guide the actor's action. Besides, target actor and critic networks are utilized to increase training stability. Different from TD3, the SRL training environment randomly assigns T_{p} value for each episode, and the training data are stored separately in several independent buffers according to T_{p}. Each supervisor samples data from its own buffer while the actor receives data from all buffers. In this manner, the supervisor avoids providing inappropriate supervisory information, and the actor can receive all supervisory information. The detailed update process of each component is illustrated as follows.

2) ACTOR

To find the optimal policy μ^* that maximizes the expected return, the loss function of the actor can be updated by taking the gradient of the expected return. According to PG, the loss function update of the actor can be written as

$$\nabla_\phi J_{\mathrm{a}}(\phi) = \mathbb{E}_{s\sim p_\mu}[\nabla_a Q_\theta(s, a)|_{a=\mu_\phi(s)} \nabla_\phi \mu_\phi(s)], \tag{21}$$

where J_a is the loss of the actor. θ is the parameter of critic, p_μ denotes the discounted state distribution for a policy μ. To avoid unreasonable actions, the output of the actor should be clipped according to constraints, namely, clip$(a, -b, b)$.

3) SUPERVISOR

With the supervisor, the loss function update of the actor in (21) should be modified. We use the mean squared error (MSE) loss to calculate supervision loss for each supervisor and sum up to form J_s,

$$\nabla_\phi J = (1-\alpha_\mathrm{s})\nabla_\phi J_\mathrm{a} + \alpha_\mathrm{s}\nabla_\phi J_\mathrm{s}, \tag{22}$$

$$J_\mathrm{s} = \sum_{l=1}^{m} \mathbb{E}_{s\sim p_\mu}[(\mu_\mathrm{sl}^l(s) - \mu_\phi(s))^2], \tag{23}$$

where J is the revised loss of the actor, J_s is the supervision loss, $\alpha_\mathrm{s} \in [0, 1]$ represents the trade-off between DRL and SL contribution, $\mu_\mathrm{sl}^l(s)$ is the lth supervisor's policy, m is the number of supervisors.

4) CRITIC

The policy update of the actor is delayed by a rate DP to let the critic has a better estimation of Q. The update of the critic is by minimizing the critic loss

$$\min J_{\mathrm{c}(\theta)} = \mathbb{E}_{r,s\sim p_\mu}[(Q_\theta(s, a) - y)^2], \tag{24}$$

where J_c is the critic loss, y is calculated by the target critic.

5) TARGET NETWORK

There are two target critics and one target actor. By using double Q learning to take the minimum Q value of the target critics, the Q value overestimation issue is mitigated. The target critics calculate

$$y = r(s, a) + \gamma \min_{k=1,2} Q_{\theta'_k}(s', \tilde{a})|_{\tilde{a}=(\mu'(s')+\mathrm{clip}(\mathcal{N}))}, \tag{25}$$

where θ' is the parameter of the target critic, $\tilde{a}$ is the action taken by the target actor, $\mu'(s')$ is the policy of the target actor at state s'. A target policy smoothing is implemented by adding a small stochastic noise to the target actor for mitigating overfitting.

Target networks are updated at regular intervals to enable more stable learning, namely, soft update. The soft update can be written as

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi', \ \theta' \leftarrow \tau\theta + (1-\tau)\theta', \tag{26}$$

where τ is the soft update rate.

The detailed training procedures are shown in algorithm 1 and algorithm 2. The hyperparameters for supervisor pre-training and agent training are the same, except for the different neural network structures. For the supervisor, both actor and critic have 256, 256, 128, 64, and 64 units for hidden layers. For the agent, both actor and critic have 400, 300, 200, 100, and 64 units for hidden layers. Each of the hidden layers is followed by a Relu activation function. The output layer of the actor is followed by a Tanh activation function, while the output layer of the critic does not have any activation function. The target networks have the same structure as the corresponding actor or critic. The inputs are normalized for all networks. The learning rate for the actor and the critic is 10^{-5} and 10^{-4}, respectively. The optimizer is Adam. $\tau = 5 \times 10^{-4}$, $\gamma = 0.99$, $DP = 2$. The training batch size is 128, and the replay buffer capacity is 2^{20}. The noise is subject to normal distribution $\mathcal{N}(0, 0.2)$ and the noise is clipped to the range $(-0.5, 0.5)$. The simulation is on Python 3.9.13 with PyTorch 1.12.1 by an RTX 3070 GPU and 32 GB RAM.

Algorithm 1 Pre-Training

```
Randomly initialize actor $\mu_\phi$ and critic $Q_{\theta_1}$, $Q_{\theta_2}$, respectively, with random weights $\phi$, $\theta_1$, and $\theta_2$
Initialize target networks $\phi'$, $\theta'_1$, and $\theta'_2$ with weights $\phi' \leftarrow \phi$, $\theta'_1 \leftarrow \theta_1$, and $\theta'_2 \leftarrow \theta_2$
Initialize buffer $B$
for episode = 1, Max do
  Receive the initial observation $s_0^\mathrm{s}$
  for $i = 1, N$ do
    Select $a \sim \mu_\phi(s) + \mathcal{N}$, clip$(a, -b, b)$, execute $a$ and observe $r$, $s'$, $b'$
    Store transition $(s, a, s', r, b, b', done)$ to $B$
    Sample a random minibatch of $N_\mathrm{m}$ transitions from $B$
    Select $\tilde{a} \sim \mu'_\phi(s') + \mathcal{N}$, clip$(\tilde{a}, -b', b')$, then calculate $y$ by (25)
    $\theta_k \leftarrow \arg\min_{\theta_k} N_\mathrm{m}^{-1} \sum(y - Q_{\theta_k}(s, a))^2$
    if $i$ mod $DP$ then
      Update $\phi$ by (21), $\phi'$, $\theta'_1$, and $\theta'_2$ by (26)
    end if
  end for
end for
```

E. MODEL VERIFICATION

After training, the IETTO model is tested by simulations to verify its model performance. The detailed simulation setup, results, and discussions are reported in Section IV. For practical application purposes, the proposed model can be deployed on onboard computers. With the received real-time information from onboard and external sensors, other trains, and control centers, the proposed model can dynamically generate optimal train trajectories online to address uncertain disturbances and rescheduled trip times.

IV. SIMULATION AND RESULT

A. SIMULATION SETUP

Simulations for three case studies are carried out to demonstrate the SRL-IETTO performance. The infrastructure, train, and line data are from an in-service subway line containing 13 sections (14 stations) and a total length of 22.73 km. The model is trained on one section of 2.63 km and evaluated on the whole line. The speed limits and gradient profile of the training section [41] are shown in Table 2. The timetable

Algorithm 2 Agent Training

```
Randomly initialize actor $\mu_\phi$ and critic $Q_{\theta_1}$, $Q_{\theta_2}$, respectively, with random weights $\phi$, $\theta_1$, and $\theta_2$
Initialize target networks $\phi'$, $\theta_1'$, and $\theta_2'$ with weights $\phi' \leftarrow \phi$, $\theta_1' \leftarrow \theta_1$, and $\theta'_2 \leftarrow \theta_2$
Initialize buffer $B_1, B_2, \ldots, B_m$, load $\mu_{\mathrm{sl}}^1, \mu_{\mathrm{sl}}^2, \ldots, \mu_{\mathrm{sl}}^m$
for episode = 1, Max do
    Receive the initial observation $s_0$ and $s_0^{\mathrm{s}}$
    for $i = 1, N$ do
        Select $a \sim \mu_\phi(s) + \mathcal{N}$, clip$(a, -b, b)$, execute $a$ and observe $r, s', b'$
        Store transition $(s, a, s', r, b, b', done)$ to $B_1, B_2, \ldots, B_m$, according to $T_{\mathrm{p}}$
        Sample a random minibatch of $N_{\mathrm{m}}$ transitions, equally from $B_1, B_2, \ldots, B_m$
        Select $\tilde{a} \sim \mu'_\phi(s') + \mathcal{N}$, clip$(\tilde{a}, -b', b')$, then calculate $y$ by (25)
        $\theta_k \leftarrow \arg\min_{\theta_k} N_{\mathrm{m}}^{-1} \sum (y - Q_{\theta_k}(s, a))^2$
        if $i$ mod $DP$ then
            Update $J_{\mathrm{s}}$ by (23), $\phi$ by (22), $\phi'$, $\theta_1'$, and $\theta_2'$ by (26)
        end if
    end for
end for
```

TABLE 2. Gradient and speed limits for training.

Item	Value	Segment (km)	Value	Segment (km)
Speed limits (km/h)	50	$[0, 0.31]$	65	$(0.64, 1.32]$
	80	$(0.31, 0.64]$	80	$(1.32, 2.63]$
Gradients (‰)	0	$[0, 0.02]$	−3	$(1.15, 1.55]$
	2	$(0.02, 0.34)$	8	$(1.55, 2.06]$
	3	$(0.34, 0.65]$	−3	$(2.06, 2.63]$
	−10.4	$(0.65, 1.15]$	–	–

and section length of the line [42] are shown in Table 3. The type of train used is DKZ32 EMU, which has six vehicle units (three are traction units). The train parameters are shown in Table 4. We set E_{lim} to be 162 J/(km·kg) based on practical and simulation data [43] of the same training section. In *Case One*, the overall model performance without disturbances or rescheduled trip times on various sections is verified. In *Case Two*, the dynamic online train trajectory optimization capability is verified with disturbances and rescheduled trip times. In *Case Three*, several algorithm parameters and their influences on model generalization capability are analyzed by comparison of learning curves.

B. CASE ONE

We compare SRL-IETTO with other approaches on multiple sections to illustrate its overall model performance without disturbances or rescheduled trip times. The following approaches are selected for comparison: 1) *manual driving* (*MD*); 2) *ATO*-generated trajectories with proportional-integral-derivative (PID) controller; 1) − 2) are the practical driving data with no departure delays of the line on March 2012. Half of the data are MD, and the others are ATO since both types of driving were used at that time. 3) *RTO algorithm*, which is a comprehensive knowledge-based system with a collection of expert knowledge rules. The selection of experts requires prior data collection, surveying, expert selection, data mining, and summarizing. Noted that RTO is unable to handle disturbances. 1) − 3) are all presented by [6]. The index E of the above algorithms is recalculated using (16) for comparison purposes. Since the calculation method of index C in 1) − 3) is different from the proposed approach, we choose 4) *STO algorithm* [9] for C comparison. STO utilized advanced DRL algorithms such as DDPG and NAF to handle continuous action space for solving the TTO. The comparison is valid because the evaluation standard should be the same for the whole line. The performance of SRL-IETTO is averaged across five runs. Five pre-trained supervisors are default used with planned trip times of 185 s, 197 s, 209 s, 221 s, and 234 s, respectively. They are equally distributed within $T_{\mathrm{p}}^{\min}$ and $T_{\mathrm{p}}^{\max}$, namely, 183 s and 234 s, respectively. Fig. 4 shows the trajectories generated by SRL-IETTO on these training sets. The trajectories are smooth and have no violations of on-road speed limits.

TABLE 3. Timetable and section length.

Station	1	2	3	4	5	6	7
Arrival time (s)	0	220	358	545	710	835	979
Dwell time (s)	30	30	30	30	35	30	30
Length (m)	2631	1275	2366	1982	993	1538	1280
Station	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Arrival time (s)	1112	1246	1440	1620	1790	1927	2087
Dwell time (s)	30	30	30	30	35	45	–
Length (m)	1354	2338	2265	2086	1286	1334	–

TABLE 4. Train parameters.

Item	Value
Train mass (kg)	2×10^5
Basic resistance force (kN)	$0.0085v^2 + 0.144v + 3.48$
Maximum traction force (kN)	$310\ (v < 36\ \mathrm{km/h})$ $310 - (v - 36) \times 20$ $(36 < v < 80\ \mathrm{km/h})$

Fig. 5 and Table 5 show overall model performance comparison with no disturbances. The upper part shows the optimal trajectories of SRL-IETTO, while the bottom part shows the results in index E and index C. The bars represent the results in index E. The circles with the dotted line and the light-shaded area represent the results of index C. The results show that SRL-IETTO can satisfy on-road speed limits and stopping accuracy in all sections. SRL-IETTO achieves the best performance on index E among all the approaches and outperforms MD in average energy saving of 18.5% across all sections. Although RTO achieves similar performance on index E as SRL-IETTO, it is unable

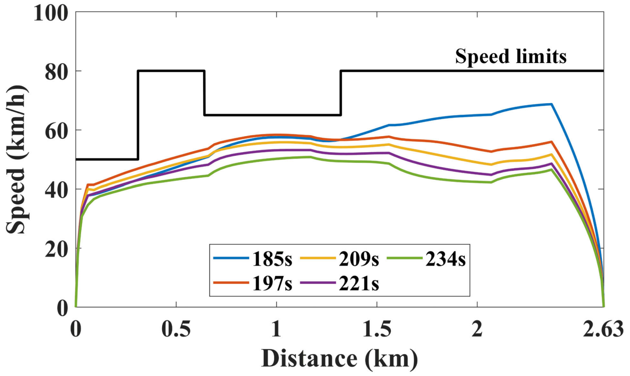


FIGURE 4. Trajectories generated on training sets.

TABLE 5. Performance across all sections.

Item	Δt (s)	E (J/(km·kg))	P_{E} (%)	C (m/s^3)	P_{C} (%)
MD	2.5 ± 2.4	147.0 ± 31.0	–	7.5–14.0	–
ATO	1.7 ± 1.6	154.7 ± 31.0	−5.2	–	–
RTO	2.0 ± 1.0	120.1 ± 20.9	18.3	–	–
STO	–	–	–	4.0–5.8	34.7
SRL-IETTO	2.2 ± 1.0	119.8 ± 23.8	18.5	3.4 ± 0.9	54.7

[1] Δp and v_{s} are always zero; "±" denotes a single standard deviation; P_{E} (%) = (average E of MD − average E of other approaches) / average E of MD × 100%; P_{C} (%) = (minimum C of MD − average C of other approaches) / minimum C of MD × 100%.

TABLE 6. Performance comparison with RL-based algorithms (without disturbances / rescheduled trip times).

Item	Δt (s)	P_{E} (%)	P_{C} (%)
MD	1 –4.2	–	–
STO	1.0	9.4 – 11.7	49.5
SRL-IETTO	1.0	24.8	47.3

[1] Δp and v_{s} are always zero.

to handle disturbances. SRL-IETTO achieves the best performance on index C compared with the practical solution and outperforms the practical solution in an average energy saving of 54.7% across all sections. Although higher than ATO, the index Δt of SRL-IETTO is still within 3 s. The variance of the trip time error of MD is huge, indicating that MD has unsatisfactory performance on punctuality in some sections.

We further compare the proposed approach with RL-based algorithms reported in the literature with no disturbances. Currently, RL-based algorithms have only been tested on several sections of the subway line we simulated. Therefore, for comparison purposes, we choose the STO algorithms and simulated section presented in [9] as an example. The simulated section line data, including speed limits and gradients, are the same as in [9]. The planned trip time is 101 s, and the section length is 1280 m. The comparison results are shown in Table 6. From the table, SRL-IETTO achieves the best performance, with an energy saving of 24.8% compared to MD. Compared to RL-based algorithms, the proposed approach increases the energy-saving rate by at least 13.1%. In terms of ride comfort, the proposed approach achieves similar performance as STO.

TABLE 7. Performance with dynamically adjusted trip times.

Item	Adjustment	Δt (s)	E (J/(km·kg))	C (m/s^3)
Scheduled	–	1.1	73.3	2.6
Scenario 1	10 s earlier	1.1	80.6	3.0
	25 s earlier	2.3	102.0	2.5
Scenario 2	10 s later	1.0	71.0	2.4
	25 s later	1.0	77.8	3.0

[1] Δp and v_{s} are always zero.

TABLE 8. Monte Carlo results.

Item	Disturbances?	Δt (s)	E (J/(km·kg))	C (m/s^3)
SRL-IETTO	–	1.6	100.7	3.0
	$\mathcal{D}^{\mathrm{bd}}$	1.6 ± 0.9	109.2 ± 13.6	3.2 ± 3.1
	$\mathcal{D}^{\mathrm{do}}$	1.9 ± 1.0	101.5 ± 11.7	2.7 ± 1.9

[1] Δp and v_{s} are always zero; "±" denotes a single standard deviation.

C. CASE TWO

We verify the dynamic online train trajectory optimization capability of the proposed model with disturbances and rescheduled trip times. First, we use Fig. 6 as an example to illustrate the model performance of the proposed model with disturbances and rescheduled trip times. Suppose the planned trip time is scheduled as 210 s for the training section. Fig. 6(a) show an accident occurs when the train runs 500 m. The train is informed at this moment to arrive at the station 10 s / 25 s earlier, respectively. This indicates that T_{p} is changed to 200 s / 185 s, respectively. A red star marker represents the position where the accident happens. Fig. 6(b) are similar, except that the accident happens when the train runs 1500 m and the train is required to arrive 10 s / 25 s later, respectively. It can be observed that when the train receives the accident information, the proposed model will change the driving strategy (action a) since the model input (state s) is changed due to the change of reserved trip time. Table 7 shows the detailed model performance. The trip time error is always within 3 s. The evaluation index E is larger than the example with scheduled T_{p}, indicating extra energy consumption due to acceleration. The index C is larger than the example with scheduled T_{p}, indicating slightly uncomfortable passengers may feel due to acceleration or deceleration.

To test the overall model performance under disturbances / rescheduled trip times, we then perform 2000 times of Monte Carlo simulations. Section 9 is between two busy stations and is suitable for demonstrating the test results. $T_{\mathrm{p}}^{\min}$ and $T_{\mathrm{p}}^{\max}$ of section 9 are 150 s and 185 s, respectively. The distributions of trip time changes are referred to [6] and [44]. $t_{\mathrm{d}}^{\mathrm{bd}}$ is subject to a Weibull distribution where the shape

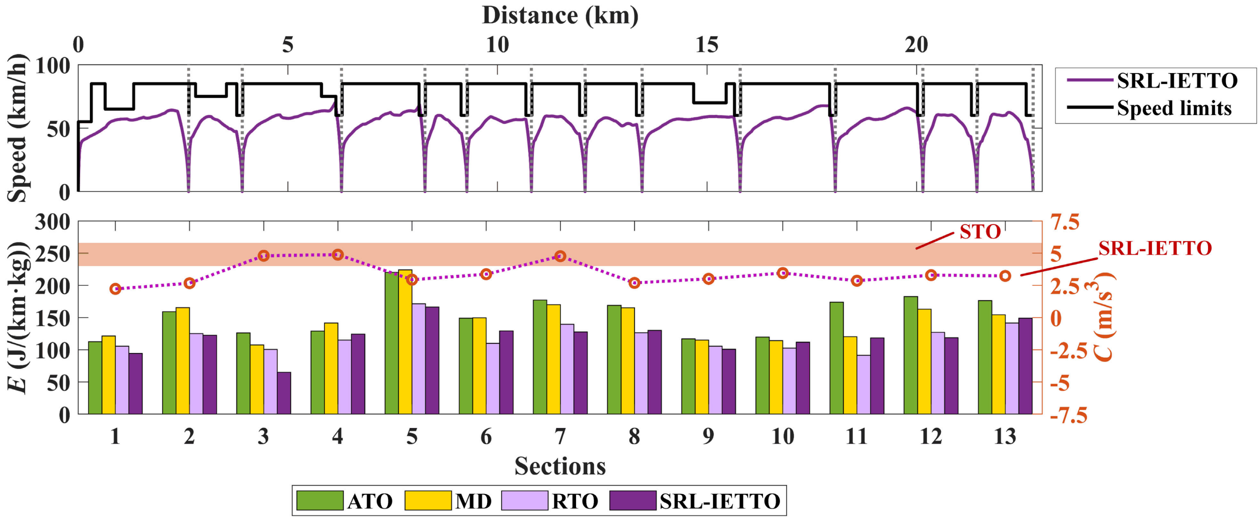
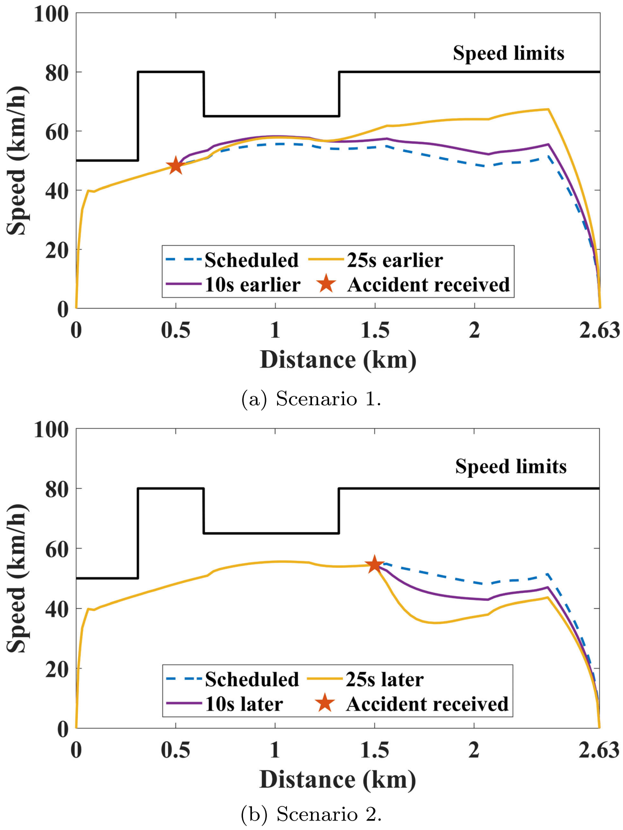


FIGURE 5. Model performance comparison across all sections.



(a) Scenario 1.

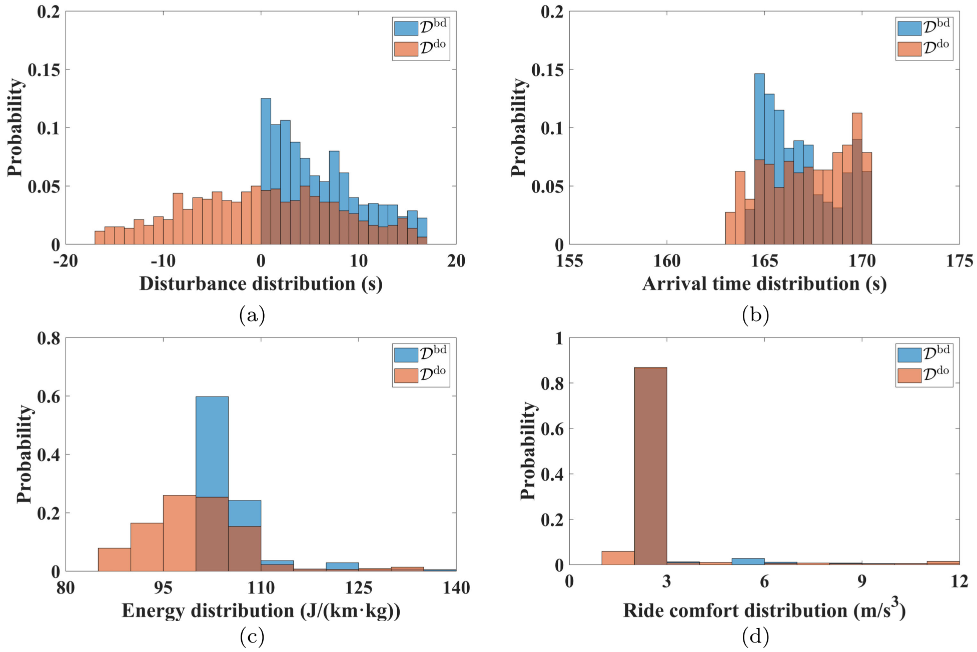
(b) Scenario 2.

FIGURE 6. Trajectories of SRL-IETTO with adjusted trip times.

parameter is 0.8, and the scale parameter is $(T_{\mathrm{p}}^{\max} - T_{\mathrm{p}}^{\min})/2$. $t_{\mathrm{d}}^{\mathrm{bd}}$ is non-negative since disturbances or rescheduling commands before departure usually cause delays. $t_{\mathrm{d}}^{\mathrm{do}}$ is subject to a Normal distribution where the mean value is 0, and the variance value is $(T_{\mathrm{p}}^{\max} - T_{\mathrm{p}}^{\min})/4$. For simulation purposes, x_{d} is set to occur within the first half of the trip. This is because when the train is close to the destination, it is difficult or even impossible to significantly change trip time by train control.

Fig. 7 shows the Monte Carlo results in histograms. The blue and red colors of the histograms represent simulations that are subject to $\mathcal{D}^{\mathrm{bd}}$ and $\mathcal{D}^{\mathrm{do}}$, respectively. Fig. 7(a) shows the distribution of $t_{\mathrm{d}}^{\mathrm{bd}}$ and $t_{\mathrm{d}}^{\mathrm{do}}$, which denotes the distribution of disturbances. Fig. 7(b) shows distribution of $(T+t_{\mathrm{d}}^{\mathrm{bd}}$ or $T+t_{\mathrm{d}}^{\mathrm{do}})$ and denotes the arrival time. Fig. 7(c) and Fig. 7(d) shows the distribution of energy and ride comfort. The average model performance of the Monte Carlo simulations is reported (see Table 8). It can be observed that disturbances vary on a broad time distribution (0-17.5 s for $\mathcal{D}^{\mathrm{bd}}$, -17.5-17.5 s for $\mathcal{D}^{\mathrm{do}}$), but the arrival time distribution is concentrated around the planned trip time (around 163-170 s). This indicates that the probability of delay is very small across all simulations (0-3.3 s for $\mathcal{D}^{\mathrm{bd}}$, 0-3.8 s for $\mathcal{D}^{\mathrm{do}}$) and the average trip time errors are within 2 s under both $\mathcal{D}^{\mathrm{do}}$ and $\mathcal{D}^{\mathrm{bd}}$. The punctuality against $\mathcal{D}^{\mathrm{do}}$ is worse than against $\mathcal{D}^{\mathrm{bd}}$. This indicates the additional trip time error caused by trajectory changes during operation. The energy distribution varies due to the extra energy consumption for acceleration and deceleration to guarantee punctuality. Most of the resulting energy consumptions are concentrated within a small range (around 85-110 (J/(km·kg)) for $\mathcal{D}^{\mathrm{bd}}$, and 100-110 (J/(km·kg)) for $\mathcal{D}^{\mathrm{do}}$) with the average energy consumption close to results without disturbances. The ride comfort distribution is similar to energy distribution, except that it is more concentrated (2-3 (m/s^3)).

The Monte Carlo simulation shows that SRL-IETTO can efficiently overcome the disturbances before departure and during operation and keeps model performance in terms of punctuality, energy saving, and ride comfort via online timetable adjustment. We then compare SRL-IETTO with



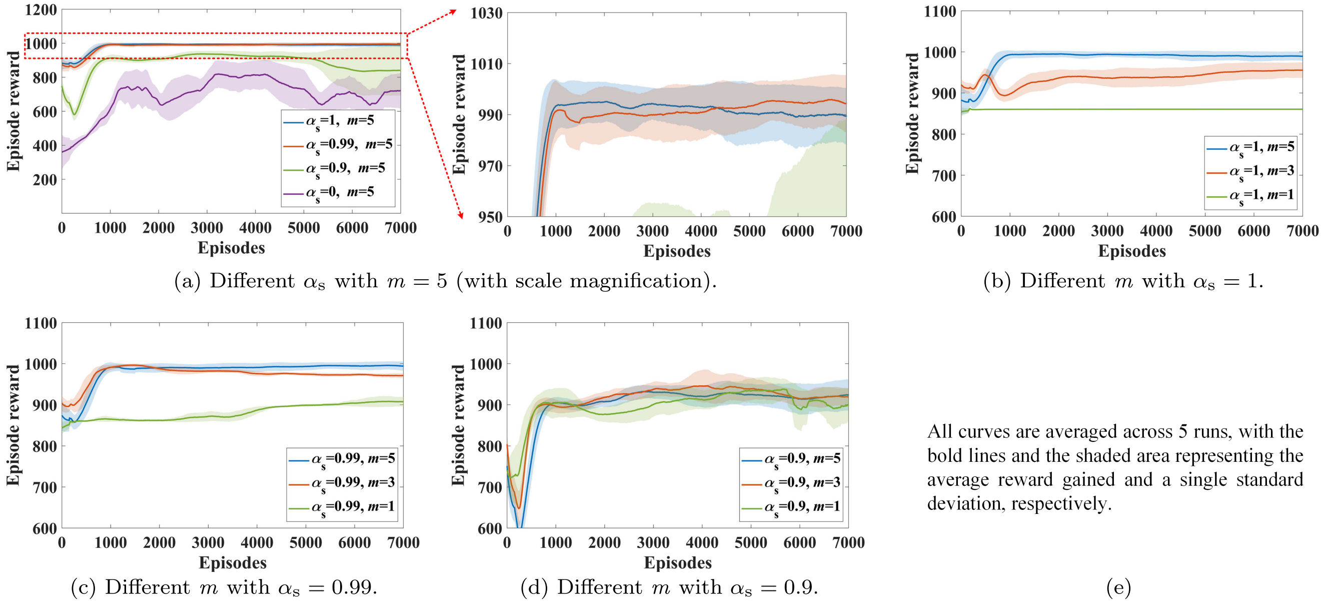
(a)

(b)

(c)

(d)

FIGURE 7. Probability distributions under the Monte Carlo simulations.



(a) Different α_{s} with $m = 5$ (with scale magnification).

(b) Different m with $\alpha_{\mathrm{s}} = 1$.

(c) Different m with $\alpha_{\mathrm{s}} = 0.99$.

(d) Different m with $\alpha_{\mathrm{s}} = 0.9$.

All curves are averaged across 5 runs, with the bold lines and the shaded area representing the average reward gained and a single standard deviation, respectively.

(e)

FIGURE 8. Reward curves under different α_{s} and m (with scale magnification).

RL-based algorithms reported in the literature that consider disturbances (See Table 9). We chose the ITO and ITOR algorithms based on Q-learning for comparison. It can be observed that SRL-IETTO reduces maximum trip time error by at least 24.0% and 11.6% against a broader disturbance range compared with ITO and ITOR, respectively. Moreover, the computational time to re-generate the optimal train trajectory after disturbances is about 0.07 s. This fast response time indicates that the SRL-IETTO can generate or re-generate the optimal train trajectory online.

D. CASE THREE

We analyze the effect of several algorithm parameters on model generalization capabilities. For discussion purposes, the following model performance is evaluated with

TABLE 9. Performance comparison with RL-based algorithms (with disturbances / rescheduled trip times).

Item	Disturbances (s)	max Δt (s)
ITO	≤ 20	5.0
ITOR	10	4.3
SRL-IETTO	≤ 35	3.3 ($\mathcal{D}^{\mathrm{bd}}$) 3.8 ($\mathcal{D}^{\mathrm{do}}$)

[T_{p} sampled every 5 s from [$T_{\mathrm{p}}^{\min}$, $T_{\mathrm{p}}^{\max}$], namely, 190 s, 195 s, …, 230 s.

1) EFFECT OF α_{s}

α_{s} is the parameter that controls the trade-off between SL and RL contributions to the actor loss (see Fig. 8(a)). When $\alpha_{\mathrm{s}} = 0$, it is pure DRL training. The rewards gained are significantly less than other curves within the maximum episode length, and the learning curve has a large variation even trained for a long time. This indicates that pure DRL training is time-consuming and more difficult to find the optimum due to the problem complexity compared with SRL. When α_{s} is larger, the introduced SL supervision accelerates training and improves average performance on different trip times as more rewards are gained. Nevertheless, from the scale magnification of Fig. 8(a), if $\alpha_{\mathrm{s}} = 1$, the curve slowly decreases after gaining a high reward. This is due to the overfitting of the agent to the supervisor's policy $\mu_{\mathrm{sl}}(s)$. Note that $\alpha_{\mathrm{s}} = 1$ does not represent pure SL training since DRL is in effect for the critic.

2) EFFECT OF m

m denotes the number of supervisors. The learning curves for different m with constant α_{s} are shown in Fig. 8(b), Fig. 8(c), and Fig. 8(d). With larger m, the rewards gained within the maximum episode length increase. The highest reward is obtained by $m = 5$. When $\alpha_{\mathrm{s}} = 0.9$, since the total SL contribution is small, the rewards gained under different m are similar. $\alpha_{\mathrm{s}} = 0.99$ and $m = 5$ is the best and default parameter for SRL-IETTO. Compared with pure DRL, the designed SRL training architecture improves model generalization capability while accelerating training.

V. CONCLUSION

In this paper, an SRL-IETTO approach is proposed for iATO in real-time train operation of modern URTNs by hybrid-integrating DRL and SL. An IETTO model is established to handle uncertain disturbances in real-time train operation and generate optimal energy-efficient train trajectories online, considering energy saving, ride comfort, punctuality, and safety. Numerical simulations are implemented to validate the effectiveness of the SRL-IETTO using in-service subway line data. The results have demonstrated the superior energy saving of the proposed approach for train trajectory optimization and provide satisfactory performance on evaluation indices of ride comfort, punctuality, and safety. The energy saving and ride comfort indices show significant improvements of 18.5% and 54.7% on average, respectively, compared to the practical driving data. The adaptability of the proposed approach to online trip time adjustments within the practical running time range has been confirmed by Monte Carlo simulations. The maximum trip time error under uncertain disturbances in real-time train operation decreased by 11.6% compared to other intelligent TTO algorithms. Future research will improve the proposed approach with more practical constraints in complex operational scenarios.

TABLE 10. Performance with different coefficient values.

r_{T}, r_{E}, r_{C}	E (J/(km·kg))	Δt (s)	v_{s}	C (m/s^3)
0.4, 0.6, 100	96.7 ± 34.9	3.5 ± 0.9	0.3 ± 0.5	12.0 ± 14.9
0.4, 0.4, 100	82.9 ± 10.9	4.2 ± 4.0	0 ± 0	2.9 ± 2.2
0.4, 0.2, 100	96.7 ± 21.2	5.4 ± 5.7	0.3 ± 0.5	10.9 ± 10.6
0.4, 0.6, 0	$\mathbf{82.5 \pm 10.2}$	$\mathbf{2.1 \pm 0.6}$	$\mathbf{0 \pm 0}$	$\mathbf{1.7 \pm 0.4}$
0.4, 0.4, 0	80.0 ± 2.6	7.5 ± 3.3	0 ± 0	2.2 ± 0.7
0.4, 0.2, 0	84.2 ± 13.0	4.1 ± 1.8	0 ± 0	4.9 ± 4.8
0.2, 0.6, 100	83.3 ± 6.4	4.7 ± 2.6	0 ± 0	3.9 ± 3.0
0.2, 0.4, 100	85.2 ± 16.0	2.8 ± 1.1	0 ± 0	7.9 ± 4.3
0.2, 0.2, 100	93.8 ± 31.5	4.9 ± 2.6	0.3 ± 0.5	13.8 ± 17.6
0.2, 0.6, 0	83.9 ± 16.5	2.6 ± 0.2	0 ± 0	4.2 ± 3.4
0.2, 0.4, 0	90.5 ± 24.7	3.0 ± 0.6	0 ± 0	10.9 ± 13.8
0.2, 0.2, 0	83.8 ± 9.5	2.3 ± 0.2	0 ± 0	2.7 ± 0.5

[1] Δp is always zero; "±" denotes a single standard deviation.

APPENDIX

The upper bound is zero for all r_{g} coefficients since we aim to minimize these reward terms. The lower bound is derived from the following equation that all coefficients must satisfy:

$$r_{\mathrm{g}_{\max}} + r_{\mathrm{T}}\Delta t + r_{\mathrm{E}}E + \frac{r_{\mathrm{C}}C}{N} - \Delta p = r_{\mathrm{g}_{\min}} > r_{\infty}/(1-\gamma), \tag{27}$$

where $r_{\mathrm{g}_{\min}}$ and $r_{\mathrm{g}_{\max}}$ are the minimum/maximum value of r_{g}, respectively.

Δp can be ignored if the maximum step N is sufficiently large that the train position differences between each step is small. Therefore, (27) can be rewritten as

$$\begin{aligned} \max \quad & r_{\mathrm{T}}, r_{\mathrm{E}}, r_{\mathrm{C}} \\ \text{s.t.} \quad & (27), 0 \leq C \leq C_{\max}, T_{\mathrm{lim}} \leq \Delta t \leq T_{\mathrm{p}}^{\mathrm{new}}, \\ & 0 \leq E \leq E_{\mathrm{lim}}, r_{\mathrm{T}}, r_{\mathrm{E}}, r_{\mathrm{C}} \leq 0, \end{aligned} \tag{28}$$

where $C_{\max} = N$ is a sufficiently large value to consider the worst ride comfort case that $|\frac{du}{dt}| > 0.3$ m/s^3 at every step.

According to Table 1, (28) is solved by the mathematical programming solver GUROBI 9.5.2 as a bilinear programming problem. The bound of r_{T}, r_{E}, and r_{C} is found: $r_{\mathrm{T}} \in [-0.43, 0]$, $r_{\mathrm{E}} \in [-0.62, 0]$, and $r_{\mathrm{C}} \in [-100, 0]$. Table 10 shows the model performance under different coefficient values, averaged by three runs on randomly selected T_{p}. The best parameters are $r_{\mathrm{T}} = 0.4$, $r_{\mathrm{E}} = 0.6$, $r_{\mathrm{C}} = 0$, with maximum trip time error within 3 s, and high energy saving and ride comfort.

REFERENCES

[1] C. Keller, F. Gluck, C. F. Gerlach, and T. Schlegel, "Investigating the potential of data science methods for sustainable public transport," *Sustainability*, vol. 14, no. 7, p. 4211, Apr. 2022.

[2] K. Huang, J. Wu, X. Yang, Z. Gao, F. Liu, and Y. Zhu, "Discrete train speed profile optimization for urban rail transit: A data-driven model and integrated algorithms based on machine learning," *J. Adv. Transp.*, vol. 2019, pp. 1–17, May 2019.

[3] S. Verkehr. (2017). *Global Urban Rail Passenger Traffic From 2005 to 2025 (in Billion Passenger-Kilometers), [Graph]*. [Online]. Available: https://www.statista.com/statistics/739801/urban-rail-passenger-transport-performance/

[4] A. Bettinelli, A. Santini, and D. Vigo, "A real-time conflict solution algorithm for the train rescheduling problem," *Transp. Res. B, Methodol.*, vol. 106, pp. 237–265, Dec. 2017.

[5] M. Volovski, E. S. Ieronymaki, C. Cao, and J. P. O'Loughlin, "Subway station dwell time prediction and user-induced delay," *Transportmetrica A, Transp. Sci.*, vol. 17, no. 4, pp. 521–539, Dec. 2021.

[6] J. Yin, D. Chen, L. Yang, T. Tang, and B. Ran, "Efficient real-time train operation algorithms with uncertain passenger demands," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2600–2612, Sep. 2015.

[7] *Railway Applications: Urban Guided Transport Management and Command/Control Systems. Part 1: System Principles and Fundamental Concepts*, Standard 62290, 2014.

[8] J. Yin, T. Tang, L. Yang, J. Xun, Y. Huang, and Z. Gao, "Research and development of automatic train operation for railway transportation systems: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 548–572, Dec. 2017.

[9] R. Zhou, S. Song, A. Xue, K. You, and H. Wu, "Smart train operation algorithms based on expert knowledge and reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 716–727, Feb. 2022.

[10] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transp. Res. B, Methodol.*, vol. 63, pp. 15–37, May 2014.

[11] G. M. Scheepmaker, R. M. Goverde, and L. G. Kroon, "Review of energy-efficient train control and timetabling," *Eur. J. Oper. Res.*, vol. 257, no. 2, pp. 355–376, 2017.

[12] K. Ichikawa, "Application of optimization theory for bounded state variable problems to the operation of train," *Bull. Jpn. Soc. Mech. Eng.*, vol. 11, no. 47, pp. 857–865, 1968.

[13] P. G. Howlett, "The optimal control of a train," *Ann. Oper. Res.*, vol. 98, nos. 1–4, pp. 65–87, 2000.

[14] A. R. Albrecht, P. G. Howlett, P. J. Pudney, and X. Vu, "Energy-efficient train control: From local convexity to global optimization and uniqueness," *Automatica*, vol. 49, no. 10, pp. 3072–3078, 2013.

[15] J. T. Haahr, D. Pisinger, and M. Sabbaghian, "A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points," *Transp. Res. B, Methodol.*, vol. 99, pp. 167–182, May 2017.

[16] P. Wang and R. M. P. Goverde, "Multiple-phase train trajectory optimization with signalling and operational constraints," *Transp. Res. C, Emerg. Technol.*, vol. 69, pp. 255–275, Aug. 2016.

[17] P. Wang, A. Trivella, R. M. P. Goverde, and F. Corman, "Train trajectory optimization for improved on-time arrival under parametric uncertainty," *Transp. Res. C, Emerg. Technol.*, vol. 119, Oct. 2020, Art. no. 102680.

[18] A. Fernández-Rodríguez, A. Fernández-Cardador, A. P. Cucala, M. Domínguez, and T. Gonsalves, "Design of robust and energy-efficient ATO speed profiles of metropolitan lines considering train load variations and delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2061–2071, Aug. 2015.

[19] X. Yang, A. Chen, B. Ning, and T. Tang, "A stochastic model for the integrated optimization on metro timetable and speed profile with uncertain train mass," *Transp. Res. B, Methodol.*, vol. 91, pp. 424–445, Sep. 2016.

[20] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman, "Integration of real-time traffic management and train control for rail networks–Part 1: Optimization problems and solution approaches," *Transp. Res. B, Methodol.*, vol. 115, pp. 41–71, Sep. 2018.

[21] X. Rao, M. Montigel, and U. Weidmann, "A new rail optimisation model by integration of traffic management and train automation," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 382–405, Oct. 2016.

[22] P. Wang and R. M. Goverde, "Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines," *Transp. Res. B, Methodol.*, vol. 105, pp. 340–361, Nov. 2017.

[23] H. Dong, H. Zhu, Y. Li, Y. Lv, S. Gao, Q. Zhang, and B. Ning, "Parallel intelligent systems for integrated high-speed railway operation control and dynamic scheduling," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3381–3389, Dec. 2018.

[24] N. Besinovic, L. De Donato, F. Flammini, R. M. P. Goverde, Z. Lin, R. Liu, S. Marrone, R. Nardone, T. Tang, and V. Vittorini, "Artificial intelligence in railway transport: Taxonomy, regulations, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14011–14024, Sep. 2022.

[25] J. Yin, D. Chen, and L. Li, "Intelligent train operation algorithms for subway by expert system and reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2561–2571, Dec. 2014.

[26] S. Ivanov, "Reinforcement learning textbook," 2022, *arXiv:2201.09746*.

[27] J. Huang, E. Zhang, J. Zhang, S. Huang, and Z. Zhong, "Deep reinforcement learning based train driving optimization," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 2375–2381.

[28] X. Meng, H. Wang, M. Lin, and Y. Zhou, "Deep reinforcement learning for energy-efficient train operation of automatic driving," in *Proc. IEEE 8th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Nov. 2020, pp. 123–126.

[29] R. Zhou and S. Song, "Optimal automatic train operation via deep reinforcement learning," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Mar. 2018, pp. 103–108.

[30] L. Pang, Y. Zhang, S. Coleman, and H. Cao, "Efficient hybrid-supervised deep reinforcement learning for person following robot," *J. Intell. Robotic Syst.*, vol. 97, no. 2, pp. 299–312, Feb. 2020.

[31] L. Wang, W. Zhang, X. He, and H. Zha, "Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Jul. 2018, pp. 2447–2456.

[32] C. Yu, G. Ren, and Y. Dong, "Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units," *BMC Med. Informat. Decis. Making*, vol. 20, no. 3, pp. 1–8, Jul. 2020.

[33] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Supervised Actor-Critic Reinforcement Learning*. Hoboken, NJ, USA: Wiley, 2004, ch. 7, pp. 359–380.

[34] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[35] L. Matignon, G. J. Laurent, and N. L. Fort-Piat, "Reward function and initial values: Better choices for accelerated goal-directed reinforcement learning," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*. Berlin, Germany: Springer, 2006, pp. 840–849.

[36] J. P. Powell and R. Palacín, "Passenger stability within moving railway vehicles: Limits on maximum longitudinal acceleration," *Urban Rail Transit*, vol. 1, no. 2, pp. 95–103, 2015.

[37] J. Yin, D. Chen, and Y. Li, "Smart train operation algorithms based on expert knowledge and ensemble CART for the electric locomotive," *Knowl.-Based Syst.*, vol. 92, pp. 78–91, Jan. 2016.

[38] S. Fujimoto, H. V. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Stockholm, Sweden, Jul. 2018, pp. 1587–1596.

[39] S. Su, X. Li, T. Tang, and Z. Gao, "A subway train timetable optimization approach based on energy-efficient operation strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 883–893, Jun. 2013.

[40] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*.

[41] C. Wang, W. Liu, Q. Tian, S. Su, and M. Zhang, "An energy-efficient train control approach based on deep Q-network methodology," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–6.

[42] X. Yang, X. Li, B. Ning, and T. Tang, "An optimisation method for train scheduling with minimum energy consumption and travel time in metro rail systems," *Transportmetrica B, Transp. Dyn.*, vol. 3, no. 2, pp. 79–98, Feb. 2015.

[43] S. Su, T. Tang, J. Xun, F. Cao, and Y. Wang, "Design of running grades for energy-efficient train regulation: A case study for Beijing Yizhuang line," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 2, pp. 189–200, Summer 2021.

[44] J. Yuan, *Stochastic Modelling of Train Delays and Delay Propagation in Stations*. Utrecht, The Netherlands: Uitgeverij Eburon, 2006.

GUANNAN LI (Student Member, IEEE) received the B.Eng. and M.Phil. degrees in electrical engineering from Wuhan University, Wuhan, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include energy management, intelligent electrified transportation, and reinforcement learning.

SIU WING OR received the B.Sc. (Hons.), M.Phil., and Ph.D. degrees in engineering physics from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 1995, 1997, and 2001, respectively. He was a Teaching Company Associate, a Research Electronic Engineer, and a Senior Research Electronic Engineer with ASM Pacific Technology Ltd., Hong Kong, from 1995 to 2001, and then a Postdoctoral Research Fellow with the Mechanical and Aerospace Engineering Department, University of California at Los Angeles, Los Angeles, CA, USA, for one and half years. He joined PolyU, as a Lecturer, in 2002. He is currently a Professor, the Director of the Smart Materials and Systems Laboratory, and the Director of the Electrical Protection and High Voltage Coordination Laboratory with the Department of Electrical Engineering, PolyU. He has authored or coauthored more than 300 publications, including two professional book chapters, more than 210 SCI journal articles, more than 100 international conference papers, in addition to the award of 43 patents. His research interests include smart materials and devices in the bulk, micro and nanoscale, the AIoT sensing and electrical condition monitoring, energy harvesting, storage and management, and electromagnetic absorption and shielding.

KA WING CHAN (Member, IEEE) received the B.Sc. (Hons.) and Ph.D. degrees in electronic and electrical engineering from the University of Bath, Bath, U.K., in 1988 and 1992, respectively. He is currently an Associate Professor and the Associate Head of the Department of Electrical Engineering with The Hong Kong Polytechnic University, Hong Kong. His general research interests include power system stability, analysis and control, power grid integration, security, resilience and optimization, and demand response management.