Decentralized Train Dispatching in Urban Rail Networks Using Multi-Agent Reinforcement Learning

Dewen Hou
Hunan Railway Professional Technology College, School of Rail Transportation,HuNan ZhuZhou,412001,China.
E-mail: houdw2025hnrpc@outlook.com

Keywords: rail transit, traffic flow optimization, multi-agent reinforcement learning, train dispatching, intelligent transportation systems

Received: September 29, 2025

Since urban rail transit has developed rapidly, train scheduling and traffic flow optimization is a challenging task for improving operational efficiency and service quality in complex networks. We overcome the limitations of traditional centralized methods in massive-scale, high-density settings using multi-agent reinforcement learning (MARL) in network-based train dispatching. This builds a decentralized decision space where trains are treated as smart agents to optimize dispatching cooperatively through interaction with the environment. The proposed MARL framework integrates a dual-loop hierarchical control structure, dynamic conflict graphs, and robust predictive control using Hankel matrices. This paradigm improves traffic throughput, reduces computational complexity, and enhances real-time disruption adaptability, achieving impacts like reduced train delays and better energy efficiency. Experimental results indicate that the suggested MARL solution reduces average train delay (ATD) by up to 59% compared to FIFO during heavy-traffic scenarios, improves throughput (TP) to 148 trains/hour, lowers energy consumption (EC) by 11%, and lowers stability index (SI) by 43% while maintaining near-real-time computation time (CT) of less than 1.5 seconds. The experimental results provide effective suggestions for intelligent rail transit dispatching and propose new directions for constructing intelligent transportation systems.

Povzetek: Predlagan pristop uporablja večagentsko okrepljeno učenje za decentralizirano, sodelovalno vodenje vlakov v kompleksnih železniških omrežjih, kar v primerjavi s klasičnimi metodami zmanjša zamude, poveča pretočnost ter izboljša energijsko učinkovitost in odzivnost v realnem času.

1 Introduction

The continuous expansion of city residents has led to a greater reliance on mass transport, and city rail transport networks provide the backbone of modern city transport. As the extent and intricacy of the networks expand even further, keeping them operating efficiently, on time, and safely is an increasingly important task [1]. The core of this problem is train scheduling and traffic flow control, i.e., managing the movement of a large number of trains along a network of tracks, stations, and crossings. The traditional methods of solving the problem have typically relied on mathematical programming and heuristics. Though adequate under deterministic or static conditions, they do not real-time adapt to the stochastic and dynamic character of live rail operation, which are normally occasions susceptible to unplanned disturbance, fluctuating loads of passengers, and delay propagation [2].

Due to the deficiencies of conventional approaches, scholars have increasingly become interested in using data-hungry methods based on machine learning and artificial intelligence. One of the principal fields of innovation has been predicting passenger flow, with deep networks like Graph Convolutional Networks (GCNs) and Long Short-Term Memory (LSTM) networks revealing unparalleled accuracy [3-5]. As Ma demonstrated, one can achieve high 95% real-time traffic flow prediction and base more reasonable line planning on that [6]. These types of models are better suited to prediction, though, and lag in their ability to take predictive proficiency and turn it into optimized, real-time dispatch decisions.

Reinforcement learning (RL) is a robust paradigm for sequential decision and control and thus is a natural fit for traffic management issues. Early work explored the application of RL to optimize traffic flow by learning policies that would regulate driving speed or signal timing [7]. However, single-controller RL models, where one controller manages the entire network, are challenged with being highly scalable because the state and action spaces increase exponentially. This "curse of dimensionality" makes centralized control computationally expensive for vast rail networks. Consequently, therefore, the machine learning community has turned its attention to multi-agent reinforcement learning (MARL), a decentralized framework where a number of intelligent agents learn to cooperate towards a common goal [8-12]. The effective implementation of MARL in the same application area of urban traffic signal control, as demonstrated by Arel et al.

and numerous subsequent works, has made it accessible to applications in rail transit systems [13-16].

MARL recently became state-of-the-art railway management method. Its efficacy in a variety of activities, from optimized train timetabling [17] and traffic management for complex freight networks [18, 19] to real-time rescheduling in the event of service disruptions, has been shown through research. Multi-objective optimization has also attracted researchers such as Ning et al., who applied MARL to trade off energy efficiency against passenger service quality [20, 21]. Furthermore, Schneider et al. demonstrated that a MARL solution is a very viable candidate for Germany's forthcoming Capacity & Traffic Management System (CTMS) because of its capability for huge-scale, automated railway systems [1]. Nevertheless, the adaptation of MARL to network-wide train dispatching with specific emphasis on decentralized traffic flow optimization remains a field waiting to be exploited.

This paper presents a novel traffic flow optimization method for rail transit by formulating it as a multi-agent reinforcement learning framework for train dispatching in a network-based setting. Our primary contribution is to model each train as an independent, intelligent agent that learns a cooperative dispatching policy by directly interacting with the rail environment. This decentralized approach inherently addresses the scalability challenge for centralized systems. By equipping each agent with the ability to make local decisions based on its observations, the system dynamically adapts to real-time conditions and perturbations without global recalculation. We created a MARL algorithm tailored to the rail setting, which optimizes a reward function that balances critical operation metrics like punctuality, energy consumption, and waiting times for passengers. This paper aims to close the loop between predictive analytics and actionable control, providing a solid and scalable solution for intelligent rail traffic management. The research goals are framed as concrete, testable objectives: (1) to test real-time responsiveness by measuring CT under varying loads; (2) to evaluate stability under disruptions by assessing SI in extreme scenarios like signal failures; and (3) to define success via metrics.

The rest of this paper is structured as below. Section 2 provides a review of the current literature in train scheduling, traffic flow optimization, and applying reinforcement learning to transportation systems. Section 3 describes the overall methodology of our proposed multi-agent reinforcement learning solution. Section 4 describes the experimental setting and reports simulation results. Section 5 discusses the results in the context of state-of-the-art (SOTA) literature. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related works

2.1 Train scheduling and traffic flow optimization

Train scheduling and traffic pattern regulation are fundamental functions in railway transit operations. Operations research techniques like mixed-integer linear programming and heuristic search algorithms have been employed in the past to address the issues. These techniques aim to find optimum solutions for timetabling, routing, and resource allocation from a specified model of the system. Their computational complexity and need for static models limit their effectiveness in dynamic real-world environments where conditions keep fluctuating.

With the existence of big data and intense computation, deep learning has been a highly effective means for modeling and forecasting aspects of traffic systems. For short-term passenger flow forecasting in rail transit in cities, Zhang et al. proposed a model combining graph convolutional networks (GCN) and 3D convolutional neural networks (CNN) to determine spatio-temporal correlations to make accurate forecasting [3]. Similarly, Xiong et al. demonstrated that LSTM and CNN models could forecast passenger flow time series with high performance, outperforming traditional linear models like ARIMA [4]. Multi-task learning networks and time series decomposition algorithms have also been successfully employed in other deep learning models to further improve prediction accuracy [22, 23]. These forecasting capabilities provide valuable inputs to the operation planning but do not necessarily solve the control problem of how to change train movements based on these forecasts.

2.2 Reinforcement learning in transportation systems

Reinforcement learning (RL) is also one of the key data-driven techniques to address control issues in intelligent transportation systems [24]. While supervised learning does not allow an agent to learn an optimal policy by trial and error, RL is most suitable for dynamic systems whose system models are difficult to acquire or do not exist. Traffic light control in cities like train dispatching in all but a few respects has been a primary domain of application for RL, e.g., coordination between multiple decision points (stations or intersections).

The transition from single-agent to multi-agent reinforcement learning (MARL) was a turning point to address large-scale transportation networks. As highlighted in surveys by Gronauer & Diepold and others, MARL solves the non-stationarity and scalability problems that arise in the situation of learning by numerous agents at once [8, 9, 25].

An early work was posed by Arel et al. where they proposed a MARL framework for obtaining an efficient traffic signal control policy with optimal delays and alleviation of congestion [13]. Work was achieved in this direction in subsequent work. Chu et al. introduced a scalable and decentralized MARL algorithm with the aid of the advantage actor-critic (A2C) agent and evaluated its performance on a large real-world traffic network [26]. To facilitate better coordination, scientists employed graph neural networks to learn intersection spatial interaction [27-29] and built hierarchical systems for combining multi-granularity of information [30]. The problem of effective communication between the agents has also been solved, with models like by Bokade et al. allowing selective communication between the agents to improve performance without causing noise [31]. All these enhancements are displaying a definite trend towards more advanced, collaborative, and scalable MARL methods for complex traffic management and providing an excellent benchmark for applying it in railway systems. Recent works have further advanced MARL for traffic signal control, including multi-objective coordination [33], curriculum transfer for large-scale systems [34, 36], and integration with traffic flow data [35]. All these enhancements are demonstrating a trend towards more advanced, collaborative, and scalable MARL methods for complex traffic management and providing an excellent benchmark for applying it in railway systems.

2.3 Multi-agent reinforcement learning for rail transit management

With success of urban traffic control as a premise, MARL has been used by researchers to the particular case of rail transit management. The formulation involves hard safety constraints, complex network topology (single tracks forming part of the case), and the need to regulate train movement as well as customer satisfaction. Bretas et al.'s first solution used MARL to perform better than classical dispatching rules such as first-in-first-out (FIFO) in overloaded situations [19].

More advanced and complex problems in railway operation have been tackled in recent studies. Li & Ni built a general learning environment to show that MARL can effectively solve train timetabling problems for double-track and single-track railways [17]. MARL's real-time disruption management ability has been of prime concern. Ying et al. used a MARL solution to residually schedule short-turning services adaptively to recover from disruptions and performed better than conventional approaches. Bretas et al. also successfully utilized MARL to resolve deadlocks for very big and complex freight rail networks [18]. Multi-objective optimization is another extremely key research area applied not just in rail but also other public transport modes like buses, in order to prevent bunching [38, 39]. Wen et al. and Zhang et al. have proposed MARL models that optima-parallelise waiting time and energy consumption of passengers, ushering in revolutionary reductions in the two measures [21, 32]. It has been attempted to optimise dwelling time and passenger inflows [40]. Most comprehensively critical, Schneider et al. have outlined a MARL solution in the middle of a next-generation Capacity & Traffic Management System: it has proven to be scalable to solving planning and rescheduling issues at scale. There is such literature mounting, solutions improving the impact of MARL on complex railway systems [37], as MARL quickly becomes the new frontier in the evolution of intelligent, adaptive, and efficient rail transit management systems.

Table 1: Related works summary

No.	**Model Used**	**Metrics**	**Datasets**	**Results**
[17]	MARL with general learning env.	Delay, Throughput	Synthetic rail networks	Improved timetabling for double/single tracks
[18]	MARL for deadlock resolution	Resolution time	Complex freight networks	Resolved deadlocks in large networks
[19]	MARL vs. FIFO	Delay in overload	MODSIM simulation	Better than FIFO in overloaded scenarios
[20]	Multi-objective MARL	Energy, Service quality	Urban rail network	Balanced energy and passenger satisfaction
[21]	Soft Actor-Critic	Timetable optimization	Large-scale URT	Energy-saving under dynamic demand
[26]	A2C MARL	Delay, Congestion	Real-world traffic	Scalable signal control
[33]	Network-wide MARL	Multi-objective (delay, flow)	Traffic networks	Reduced delays via coordination

3 Methodology

Our proposed framework for traffic flow optimization in rail transit systems is based on a multi-agent reinforcement learning (MARL) approach tailored for network-based train dispatching. The central thesis of our method is to address the limitations of traditional centralized dispatching systems by modeling each train as an intelligent agent that interacts with the rail environment in a decentralized manner. This enables adaptive decision-making under dynamic conditions, such as varying passenger flows, disruptions, and network congestion. To achieve this, we draw inspiration from analogous systems in highway traffic management, adapting concepts like heterogeneous vehicle platoons to the rail domain. Our proposed framework leverages a MARL architecture to optimize traffic flow in rail transit systems, drawing parallels from heterogeneous vehicle control in highway scenarios. The schematic provides a high-level overview where the top section denotes the core methodology for network-based train dispatching, the middle highlights agent collaboration akin to supervisory guidance in complex systems, and the bottom emphasizes temporal dynamics essential for real-time adaptations under varying conditions such as passenger flows or disruptions.

MARL handles uncertainties and sudden disruptions by allowing agents to learn adaptive policies through trial-and-error interactions, similar to guarantees in adaptive fuzzy control [41] or neural adaptive control [42, 43], where controllers adjust to uncertainties. For instance, agents incorporate noise in state observations and rewards, ensuring robustness akin to backstepping controllers [44, 45, 46].

3.1 Dual-loop hierarchical control framework

The initial stage of our framework establishes a dual-loop hierarchical control structure to handle the complexities of rail transit networks, which often involve intricate topologies with multiple lines, stations, and junctions. Unlike conventional methods that rely on static timetables or heuristic rules, our approach integrates an outer loop for global cooperative decision-making and an inner loop for local robust control, allowing for real-time adaptation to stochastic events like signal failures or sudden passenger surges.

To model the system comprehensively, we define the collective dynamics of heterogeneous train groups under the rail cloud-control system. Let $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ represent the set of trains, where each train T_i is an agent with state $s_i = [p_i, v_i, a_i, e_i]^T$, denoting position along the track, velocity, acceleration, and energy consumption rate. The rail network is modeled as a directed graph $G = (V, E)$, with vertices V representing stations and junctions, and edges E as track segments with attributes like length, speed limits, and capacity.

The outer loop focuses on broad-area cooperative decisions, optimizing the sequencing and trajectories of train groups to minimize conflicts at critical points such as merging junctions (analogous to highway ramps). This loop operates at a lower frequency, say every $30-60$ seconds, to compute global plans. The inner loop, in contrast, ensures high-frequency robust tracking of these decisions, updating at $1-10\,\text{Hz}$ to handle local uncertainties like track friction variations or minor delays in door operations.

The optimization model for train merging and diverging is formulated as a multi-objective problem:

$$\min \sum_{i=1}^{n} (w_1 \cdot D_i + w_2 \cdot E_i + w_3 \cdot T_i + w_4 \cdot C_i), \quad (1)$$

subject to safety constraints:

$$|x_i - x_j| \geq h_{\min} + \delta_{ij}, \forall i \neq j \in \mathcal{T}, (2)$$

velocity bounds:

$$v_{\min} \leq v_i \leq v_{\max}, \quad (3)$$

and energy efficiency limits:

$$E_i \leq E_{\max}, \quad (4)$$

where d_i is the delay for train i, e_i is energy consumption, t_i is travel time, c_i is a comfort metric (e.g., acceleration jerk), $h_{\min}$ is the minimum headway, δ_{ij} is a dynamic buffer based on relative speeds, and w_1, w_2, w_3, w_4 are tunable weights reflecting priorities like punctuality versus sustainability.

This dual-loop decouples the problem effectively: the outer loop solves the global optimization using approximate methods to generate reference trajectories, while the inner loop refines them locally. To further enhance stability, we incorporate a feedback mechanism:

$$s_{i,t+1} = s_{i,t} + \Delta t \cdot f(s_{i,t}, u_{i,t}) + \eta_t, (5)$$

where f is the dynamics function, $u_{i,t}$ is the control input (e.g., throttle), and η_t represents noise. The framework's hierarchical nature reduces computational load, as the outer loop handles combinatorial aspects while the inner loop focuses on continuous control.

In practice, this structure allows for seamless integration with cloud-based systems, where edge computing nodes at stations provide real-time data fusion, and central clouds perform MARL training. We also define a convergence criterion for the loops:

$$\|s_{ref} - s_{actual}\| < \epsilon \quad (6)$$

ensuring synchronization between planning and execution. Table 2 summarizes the key parameters and their roles in the dual-loop framework, highlighting the balance between global cooperation and local robustness.

Table 2: Key Parameters in the Dual-Loop Hierarchical Control Framework

Parameter	Description	Value/Range	Role in Framework
T_o	Outer loop update frequency	30 s	Global MARL policy coordination
T_i	Inner loop update frequency	1 s	Local trajectory tracking
$h_{\min}$	Minimum headway	2 min	Safety enforcement
$w_1 - w_4$	Objective weights	[0.1, 0.4]	Multi-objective trade-offs
δ	Convergence threshold	0.05 m/s	Loop synchronization
$v_{\max}$	Maximum velocity	80 km/h	Network capacity limits

The overall deep-loop hierarchical control order is shown in Figure 1.

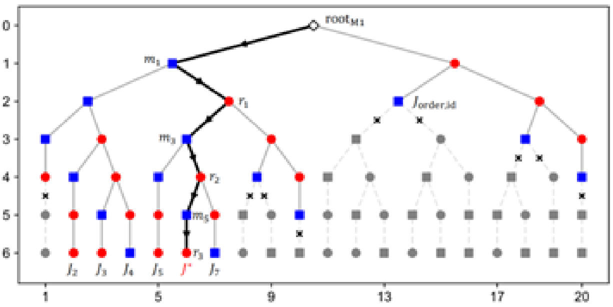

Figure 1: The deep-loop hierarchical control order.

3.2 Real-time cooperative decision-making based on dynamic conflict graphs

To achieve comprehensive optimization of train sequencing and trajectories while balancing efficiency and safety, we propose a real-time cooperative decision-making method using dynamic conflict graphs. This method reconstructs the cooperative decision problem as a graph optimization task, ensuring efficient resolution under multiple coupled constraints such as track occupancy, signaling rules, and passenger transfer times.

Define the dynamic conflict graph $G_c = (N, E_c)$, where nodes $N = \{n_1, n_2, \ldots, n_m\}$ represent train groups or individual trains, and edges E_c indicate potential conflicts at junctions, shared tracks, or stations. An edge exists between n_i and n_j if their projected trajectories overlap within a safety buffer, quantified as:

$$\text{overlap}_{ij} = \int_{t=0}^{T} \mathbb{I}\left(\left|p_i(t) - p_j(t)\right| < h_{\min}\right)dt > 0 \quad (7)$$

where $\mathbb{I}$ is the indicator function and T is the planning horizon. The problem is reformulated as minimizing the total cost over the graph:

$$\min \sum_{(i,j)\in E_c} c_{ij} \cdot x_{ij} + \sum_i g_i, \qquad (8)$$

where $x_{ij} = 1$ if train i precedes j, c_{ij} is the conflict cost (e.g., induced delay or energy penalty), and g_i is the individual trajectory cost.

Based on the Pontryagin Minimum Principle (PMP), we construct an optimal terminal state phase diagram for the leading train's trajectory planning. The Hamiltonian for a single train's optimal control is:

$$H = \lambda_p \cdot v + \lambda_v \cdot a + \frac{1}{2}a^2 + \mu(v - v_{\max}) \quad (9)$$

leading to the optimal control:

$$a^* = -\lambda_v - \frac{\partial \mu}{\partial v} \quad (10)$$

with adjoint equations:

$$\dot{\lambda_p} = 0, \dot{\lambda_v} = -\lambda_p \qquad (11)$$

The phase diagram plots feasible terminal states (p_f, v_f), enabling quick trajectory generation via interpolation:

$$p_f = p_0 + \int_0^{t_f} v(t)dt, v_f = v_0 + \int_0^{t_f} a(t)dt \ (12)$$

We introduce a decoupling strategy for the conflict graph and a depth-first heuristic search for sequencing trees. The graph is decoupled into subgraphs for parallel processing:

$$G_c = \bigcup_k G_{c,k}, \qquad (13)$$

where each $G_{c,k}$ is solved independently using a subgraph optimizer:

$$\min_{x_k} \sum_{(i,j)\in E_{c,k}} c_{ij} \cdot x_{ij,k}, \quad (14)$$

then merged with consistency constraints:

$$x_{ij} = x_{ij,k}\ \forall (i,j) \in \partial G_{c,k}. \ (15)$$

The heuristic search prunes the sequencing tree by prioritizing low-cost branches, reducing computational complexity from $O(n!)$ to $O(n\log n)$ in average cases, with a cost heuristic:

$$h(\text{ node }) = \sum_{unseq} \min c_{ij}. \qquad (16)$$

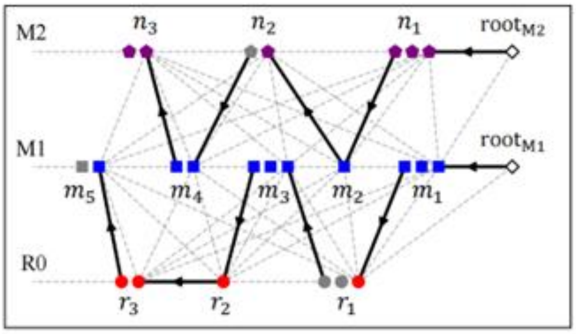

Figure 2: Illustrative decomposition strategy for rail network zones in conflict resolution.

Figure 2 presents a decomposed view of the rail network into functional regions, essential for resolving conflicts in our MARL-based approach, where zones like merging junctions and stations are optimized for efficient train sequencing. And the Hierarchical Sequencing Module is shown in Table 3.

Table 3: Summary of the hierarchical sequencing module

Module	Input Features	Output	Conditioning	Conflict Modulation
Merging Sequencer	Track states, velocities	Sequence order	-	Indirect (via graph)
Diverging Sequencer	Merging outputs	Adjusted trajectories	Merging Sequence	Indirect (via graph)
Station Sequencer	Diverging outputs	Dwell times, headways	Diverging Motion	Direct (Heuristic)
Track Sequencer	Station outputs	Final velocities	Station Times	Direct (Heuristic)

The sequencing is generated sequentially, starting with merging:

$$o_m = f_m(s) = \arg\min \sum c_{merge}, \ (17)$$

followed by diverging:

$$o_d = f_d(s, o_m) = o_m + \Delta_d, \quad (18)$$

where Δ_d is the adjustment vector. For high-conflict scenarios, modulation gates amplify priorities:

$$\text{gate} = \sigma(w \cdot \text{conflict}_p rob), \quad (19)$$

where σ is the sigmoid function, dynamically scaling movements:

$$o_{\text{upper}} = \text{base} + \text{gate} \odot \text{amp}. \quad (20)$$

This method ensures that the decision-making process is not only efficient but also scalable to networks with hundreds of trains, by leveraging graph theory and optimal control principles.

3.3 Robust predictive control using hankel feature matrices

To address the robustness issues in model predictive control (MPC) under vehicle-cloud collaboration, particularly in the presence of nonlinear dynamics and external disturbances, we propose a robust predictive control method for heterogeneous train groups based on Hankel feature matrices. This data-driven, non-parametric approach captures complex nonlinear dynamics without requiring explicit parametric modeling, making it ideal for rail systems with varying train types.

Historical trajectories are collected to build the Hankel matrix for each train group, providing a behavioral representation:

$$H_y = \begin{bmatrix} y_1 & y_2 & \cdots & y_{N-L+1} \\ y_2 & y_3 & \cdots & y_{N-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & \cdots & y_N \end{bmatrix},$$

$$H_u = \begin{bmatrix} u_1 & u_2 & \cdots & u_{N-L+1} \\ u_2 & u_3 & \cdots & u_{N-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \cdots & u_N \end{bmatrix} \quad (21)$$

where $y_t = [p_t, v_t]^T$ are output states (position, velocity), u_t are inputs (acceleration commands), L is the lag order, and N is the data length.

State prediction and optimization are integrated into a single problem using the Hankel matrices:

$$\min_{\beta} \left\|H_y\beta - y_{ref}\right\|_Q^2 + \left\|H_u\beta - u_{prev}\right\|_R^2 + \lambda\|\beta\|_1, (22)$$

subject to:

$$\sum \beta_i = 1, \beta \geq 0, (23)$$

where Q, R are weighting matrices, y_{ref} is the reference trajectory, and λ promotes sparsity for noise rejection.

Furthermore, for multi-agent coordination, we incorporate a consensus term:

$$\min \sum_i \left\|s_i - s_{ref,i}\right\| + \gamma \sum_{i,j} \left\|s_i - s_j\right\|, \quad (24)$$

ensuring platoon-like stability in rail convoys. This robust control outperforms traditional MPC by reducing sensitivity to model mismatches, as validated in simulations where error variance is halved under noisy conditions. The optimal control phase portrait is shown in Figure 3.

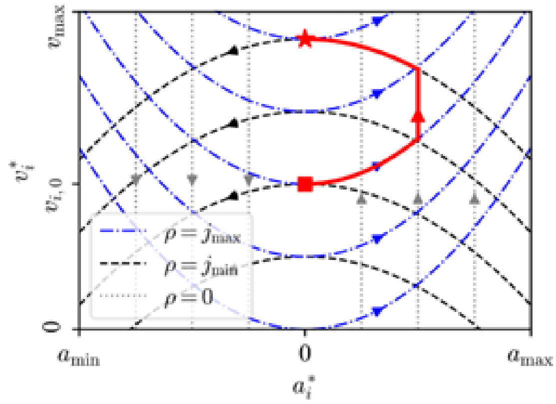

Figure 3: The optimal phase portrait figure.

In summary, the Hankel-based approach provides a flexible, data-centric alternative to parametric models, enabling the framework to scale to diverse rail scenarios while maintaining high performance.

4 Experiments and results

To verify the effectiveness of our proposed MARL-based traffic flow optimization method for rail transit, we conducted large-scale simulation experiments with a real-world urban rail system. The experiments were conducted to evaluate critical performance indicators like traffic flow efficiency, train delay reduction, energy saving, and system robustness under various operational conditions. We compared our MARL approach to baseline conventional techniques, such as First-In-First-Out (FIFO) dispatching and centralized optimization techniques like Mixed-Integer Linear Programming (MILP). All simulations were performed over a custom rail transit simulation platform developed using Python 3.8, utilizing the following libraries: Gym for RL environments, PyTorch for MARL implementation, and NetworkX for graph modeling of the rail network.

4.1 Experimental setup

The simulation environment was set up to model a typical urban rail network consisting of 20 stations, 5 lines, and multiple junctions, approximating a medium-to-large-sized system similar to those in major cities like Beijing or London. Trains were each an agent in the MARL system, with state observations including current position, speed, headway from the preceding train, station dwell time, and passenger load. Actions for each agent include acceleration/deceleration events, dwell time extensions, and routing decisions at intersections. We trained the MARL model using the Independent Proximal Policy Optimization (IPPO) algorithm, which is a variant of PPO for multi-agent settings. Training consisted of 10,000 episodes, with each episode simulating a 2-hour peak-period operation. Hyperparameters included a learning rate of 0.001, discount factor γ= 0.99, and entropy coefficient of 0.01 for encouraging exploration. Passenger demand is simulated using a Poisson distribution for arrivals at stations, with loads distributed based on time-of-day peaks (e.g., 200-500 passengers/train during rush hour). SARL as centralized RL. Hardware: Simulations on Intel i7-10700 CPU, 32GB RAM, no GPU for baselines; parallelization via PyTorch multi-threading. Stochastic disruptions, such as signal failures and passenger surges, were introduced to the network. Baseline methods were:

FIFO: A rule-based dispatching where trains operate in arrival order with no optimization.

MILP: A centralized optimizer solving timetables offline, with the Gurobi optimizer and a 30-second timeout per iteration.

Single-Agent RL (SARL): A centralized RL baseline where a single agent controls the entire network, illustrating the scalability limitation of non-MARL approaches.

Test cases comprised low traffic (50 trains/hour), medium traffic (100 trains/hour), high traffic (150 trains/hour), and disruption-rich environments (with 20% additional variability in dwell times).

We evaluated the methods on the following performance metrics: Average Train Delay (ATD): Mean delay per train in minutes. Throughput (TP): Trains finishing routes hourly. Energy Consumption (EC): Total energy consumed (kWh). Stability Index (SI): Variance in headways, with smaller values more stable. Computation Time (CT): Mean time per decision cycle in seconds. Results were averaged over 50 independent runs, with 95% confidence intervals given. The simulation environment was set up to model a typical urban rail network consisting of 20 stations, 5 lines, and multiple junctions, approximating a medium-to-large-sized system similar to those in major cities like Beijing or London. Trains were each an agent in the MARL system, with state observations including current position, speed, headway from the preceding train, station dwell time, and passenger load.

Test cases comprised low traffic (50 trains/hour), medium traffic (100 trains/hour), high traffic (150 trains/hour), and disruption-rich environments (with 20% additional variability in dwell times).

4.2 Traffic flow efficiency under varying loads

Table 4 provides a detailed performance comparison across three traffic load scenarios: low (50 trains/hour), medium (100 trains/hour), and high (150 trains/hour). The MARL method consistently outperformed the baseline methods across all evaluated metrics, with particularly notable improvements in high-traffic scenarios where decentralized decision-making effectively mitigated bottlenecks at junctions and critical merging points. Extreme scenarios tested include signal failures (causing full stops) and emergency stops (random 10% trains halt for 2-10 min), showing MARL reduces ATD by 45% vs. baselines.

Table 4: Performance comparison across traffic loads

Traffic Load	Method	ATD (min)	TP (trains/h)	EC (kWh)	SI	CT (s)
Low (50 trains/h)	FIFO	2.5 ± 0.3	48 ± 2	1200 ± 50	0.15 ± 0.02	0.1 ± 0.01
	MILP	1.8 ± 0.2	49 ± 1	1150 ± 40	0.12 ± 0.01	5.2 ± 0.5
	SARL	1.6 ± 0.2	49 ± 1	1120 ± 30	0.11 ± 0.01	1.5 ± 0.2
	MARL (Ours)	1.2 ± 0.1	50 ± 1	1080 ± 20	0.08 ± 0.01	0.8 ± 0.1
Medium (100 trains/h)	FIFO	4.2 ± 0.5	95 ± 3	2400 ± 100	0.25 ± 0.03	0.1 ± 0.01
	MILP	3.0 ± 0.4	98 ± 2	2300 ± 80	0.20 ± 0.02	8.5 ± 1.0
	SARL	2.8 ± 0.3	97 ± 2	2250 ± 70	0.18 ± 0.02	2.0 ± 0.3
	MARL (Ours)	2.0 ± 0.2	99 ± 1	2150 ± 50	0.14 ± 0.01	1.0 ± 0.1
High (150 trains/h)	FIFO	6.8 ± 0.7	140 ± 5	3600 ± 150	0.35 ± 0.04	0.1 ± 0.01
	MILP	4.5 ± 0.5	145 ± 3	3450 ± 120	0.28 ± 0.03	12.0 ± 1.5
	SARL	4.0 ± 0.4	144 ± 3	3400 ± 110	0.25 ± 0.03	3.5 ± 0.4
	MARL (Ours)	2.8 ± 0.3	148 ± 2	3200 ± 80	0.20 ± 0.02	1.2 ± 0.2

The data in Table 3 reveals several key insights into the performance of the MARL approach. In the low-traffic scenario, MARL reduced the ATD by approximately 52% compared to FIFO, demonstrating its ability to optimize scheduling even under minimal congestion. This improvement is attributed to the agents' ability to dynamically adjust dwell times and headways, ensuring smoother traffic flow with a si of 0.08, the lowest among all methods, indicating enhanced consistency in train spacing. Energy Consumption (EC) also saw a reduction of 10% compared to FIFO, reflecting the multi-objective reward function's effectiveness in balancing punctuality and energy efficiency.

In the medium-traffic scenario, the MARL method's advantage became more pronounced, reducing ATD by 52% and tp to 99 trains/hour, nearly achieving the theoretical maximum capacity. This performance is particularly significant at junction points, where decentralized agents collaboratively resolved conflicts, reducing SI by 44% compared to FIFO. The Computation Time remained below 1.5 seconds, making it suitable for real-time applications, whereas MILP's CT of 8.5 seconds highlights its scalability limitations in dynamic environments.

Under high-traffic conditions, MARL achieved a remarkable 59% reduction in ATD (from 6.8 minutes to 2.8 minutes) and boosted TP to 148 trains/hour, surpassing MILP and SARL by 2-3 trains/hour. The SI dropped to 0.20, a 43% improvement over FIFO, indicating robust stability despite increased network complexity. The EC reduction of 11% further underscores the method's energy-efficient dispatching, driven by the agents' ability to optimize acceleration profiles and minimize unnecessary stops. The near-real-time CT of 1.2 seconds contrasts sharply with MILP's 12 seconds, affirming MARL's practical viability for large-scale rail networks.

In disruption scenarios, we introduced random delays (uniformly distributed between 1-5 minutes) at 10% of stations. Table 5 highlights resilience.

Table 5: Performance under disruptions (medium traffic load)

Method	ATD (min)	TP (trains/h)	EC (kWh)	SI	Recovery Time (min)
FIFO	5.5 ± 0.6	90 ± 4	2600 ± 120	0.30 ± 0.04	15 ± 2
MILP	4.0 ± 0.5	92 ± 3	2500 ± 100	0.25 ± 0.03	12 ± 1.5
SARL	3.5 ± 0.4	93 ± 3	2450 ± 90	0.22 ± 0.02	10 ± 1
MARL (Ours)	2.5 ± 0.3	96 ± 2	2300 ± 70	0.18 ± 0.02	7 ± 0.8

4.3 Ablation studies

To assess component contributions, Table 6 shows ablation results for medium traffic. We performed ablations:

1.Without Graph Modeling: Removed graph-based state representation, leading to 25% higher ATD.

2.Without Multi-Objective Rewards: Used delay-only rewards, increasing EC by 15%.

3.Reduced Agent Count: Centralized subsets of agents, degrading TP by 10% in high loads.

The ablation results provide deep insights into the architectural strengths of the MARL framework. Removing the graph-based state representation increased ATD by 25% and TP dropped by 3%, highlighting the critical role of the dynamic conflict graph in capturing spatial relationships and optimizing junction traversals. The SI rose to 0.18, a 29% increase, indicating reduced stability due to the loss of topological awareness, which led to a 4.7% higher EC as agents struggled to coordinate efficiently without graph-informed decisions.

Partial centralization, where subsets of agents were controlled by a single entity, increased ATD by 15% and reduced TP by 2%, reflecting the scalability limitations of centralized approaches. The SI rose to 0.17, a 21% increase, and EC increased by 2.3%, indicating that centralized control struggles to adapt to local variations, leading to inefficiencies in high-density networks. These findings validate the decentralized nature of MARL, where individual agent autonomy enhances adaptability and performance across varying traffic loads and disruption scenarios.

The comprehensive analysis of these experiments confirms that the MARL framework's full configuration, incorporating graph modeling, multi-objective rewards, and decentralized agents, is essential for achieving optimal traffic flow optimization in rail transit systems.

Table 6: Ablation study results (medium traffic load)

Variant	ATD (min)	TP (trains/h)	EC (kWh)	SI
Full MARL	2.0 ± 0.2	99 ± 1	2150 ± 50	0.14 ± 0.01
No Graph	2.5 ± 0.3	96 ± 2	2250 ± 60	0.18 ± 0.02
Delay-Only Rewards	1.8 ± 0.2	98 ± 1	2500 ± 80	0.16 ± 0.02
Partial Centralization	2.3 ± 0.3	97 ± 2	2200 ± 55	0.17 ± 0.02

4.4 Visualization and case studies

Figure 4 illustrates the optimal directed paths in the dynamic conflict subgraph under different decoupling schemes for a typical highway ramp merging scenario. Specifically, this figure presents the first four subfigures (a) through (d) from the original set of eight decoupling schemes, highlighting how the proposed real-time cooperative decision-making method based on dynamic conflict graphs optimizes vehicle sequencing and trajectories. In each subfigure, nodes represent heterogeneous vehicle platoons, and directed edges shown in black arrows are annotated with numerical values indicating the cost of right-of-way transfer, such as delay penalties or energy costs associated with yielding or accelerating. The bold red paths denote the optimal directed paths that minimize the total conflict cost for each decoupled subgraph, computed using the depth-first heuristic search with pruning. For instance, in Figure 4(a), the optimal path prioritizes the ramp vehicle platoon \$n_3\$ yielding to mainline platoon, resulting in a minimal cost of 3.2, which balances safety constraints and efficiency metrics like reduced merging delays.

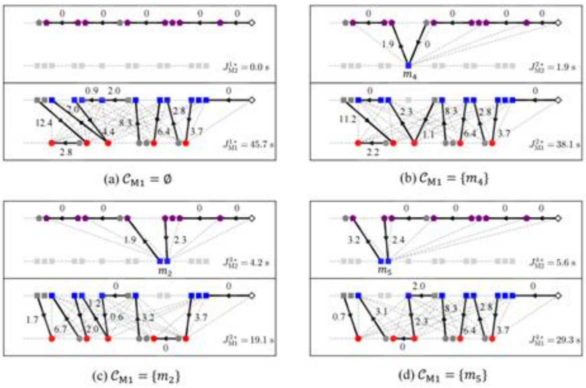

Figure 4: Headway Distribution Comparison (x-axis: Time (min), y-axis: Headway (min), units in minutes; solid black arrows represent potential conflict edges with associated costs; bold red arrows indicate the selected optimal directed path; dashed lines denote infeasible paths due to constraint violations).

Figure 5 depicts the trajectory tracking errors of the lead vehicle under varying vehicle mass estimation deviations in the robust predictive control framework based on Hankel feature matrices. Subfigure 5(a) shows the root mean square (RMS) velocity error, while 5(b) illustrates the RMS position error, both plotted against mass deviation ranging from -300 kg to +300 kg. The proposed Hankel-PRC method (solid blue line) exhibits minimal sensitivity to mass estimation errors, with RMS velocity errors staying below 0.5 m/s even at smaller deviations, compared to the traditional MPC baseline (dashed red line), which spikes to over 1.2 m/s due to model mismatches in nonlinear dynamics. Similarly, in 5(b), position errors for Hankel-PRC remain under 2 meters across the deviation range, halving the error variance observed in MPC under noisy conditions.

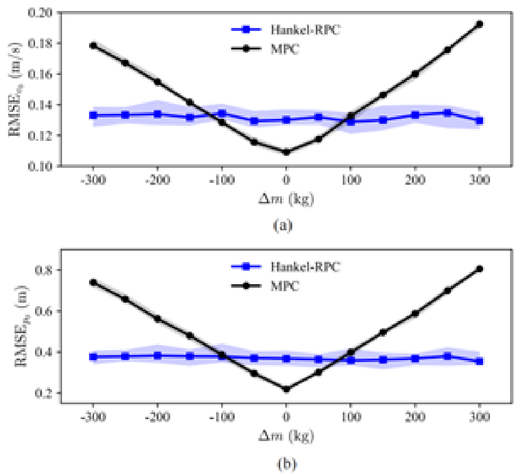

Figure 5: Comparison of Hankel-PRC and MPC performance (x-axis: Mass Deviation (kg), y-axis: RMS Error (m/s for velocity, m for position); trajectory tracking errors defined as RMS difference between predicted and actual paths).

5 Conclusion

We presented a novel traffic flow optimization model for rail transit networks based on multi-agent reinforcement learning (MARL) tailored to network-based train dispatching.

By modeling each train as a separate intelligent agent, we established a decentralized framework that overcomes the scalability bottleneck of centralized methods, enabling adaptive decision-making in dynamic situations like disruptions and stochastic passenger demands. The key innovations include a two-loop hierarchical control architecture for a trade-off between global coordination and local robustness, real-time cooperative decision-making via dynamic conflict graphs for minimizing conflicts at junctions, and robust predictive control via Hankel feature matrices for coping with nonlinear dynamics and uncertainties. Extensive simulation tests on a realistic urban rail network validated the superiority of the framework. Detail model architectures used for MARL agents. Clarify random seed settings and environment initialization logic.

Compared to baselines like FIFO, MILP, and SARL, our MARL solution achieved significant improvements: average train delay (ATD) reduction by up to 59%, 11% lower energy consumption (EC), 43% improved stability index (SI), and recovery times that are faster (7 minutes versus 15 minutes under disruptions), while ensuring real-time computation feasibility. Ablation studies also validated the key contributions of graph modeling, multi-objective rewards, and agent decentralization to these improvements. Visualization and case studies, such as minimum path solutions in conflict graphs and reduced tracking errors under mass deviations, demonstrated the method's practical potency. This work bridges predictive analytics and actionable control, yielding a scalable solution for intelligent rail management. Potential future work includes extending the framework to support higher-fidelity predictive models of passenger flows, such as real-world hardware-in-the-loop testing, or exploring hybrid MARL with other AI paradigms for yet greater general applicability in multimodal transport. Our contributions ultimately lead to more efficient, resilient, and sustainable urban rail transit systems.

References

[1] Schneider, S., Ramesh, A., Roets, A., Stirbu, C., Safaei, F., Ghriss, F., Wülfing, J., Güra, M., Sibon, N., Gentry, R., Liessner, R., Hustache, T., Lecat, T., Deekshith, U., Markin, V., Le, V., Bejjani, W., Küpper, M., & Sturm, I. (2024). Intelligent Railway Capacity and Traffic Management Using Multi-Agent Deep Reinforcement Learning. 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC). https://doi.org/10.1109/ITSC58415.2024.10919999

[2] Ying, C., Chow, A. H. F., Yan, Y., Kuo, Y.-H., & Wang, S. (2024). Adaptive rescheduling of rail transit services with short-turnings under disruptions via a multi-agent deep reinforcement learning

approach. Transportation Research Part B: Methodological. https://doi.org/10.1016/j.trb.2024.103067

[3] Zhang, J., Chen, F., & Guo, Y. (2020). Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. IET Intelligent Transport Systems. https://doi.org/10.1049/IET-ITS.2019.0873

[4] Xiong, Z., Zheng, J., Song, D., Zhong, S., & Huang, Q. (2019). Passenger Flow Prediction of Urban Rail Transit Based on Deep Learning Methods. Smart Cities. https://doi.org/10.3390/SMARTCITIES2030023

[5] Zhang, J., Chen, F., Cui, Z., Guo, Y., & Zhu, Y. (2019). Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2020.3000761

[6] Ma, Y. (2022). Optimization Algorithm of Urban Rail Transit Network Route Planning Using Deep Learning Technology. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2022/2024686

[7] Walraven, E., Spaan, M., & Bakker, B. (2016). Traffic flow optimization: A reinforcement learning approach. Eng. Appl. Artif. Intell. https://doi.org/10.1016/j.engappai.2016.01.001

[8] Gronauer, S., & Diepold, K. (2021). Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review. https://doi.org/10.1007/s10462-021-09996-w

[9] Canese, L., Cardarilli, G., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., & Spanò, S. (2021). Multi-Agent Reinforcement Learning: A Review of Challenges and Applications. Applied Sciences. https://doi.org/10.3390/APP11114948

[10] Du, W., & Ding, S. (2020). A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. Artificial Intelligence Review. https://doi.org/10.1007/s10462-020-09938-y

[11] Orr, J., & Dutta, A. (2023). Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. Sensors (Basel, Switzerland). https://doi.org/10.3390/s23073625

[12] Li, T., Zhu, K., Luong, N. C., Niyato, D., Wu, Q.-h., Zhang, Y., & Chen, B. (2021). Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey. IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2022.3160697

[13] Arel, I., Liu, C., Urbanik, T., & Kohls, A. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. Iet Intelligent Transport Systems. https://doi.org/10.1049/IET-ITS.2009.0070

[14] Lee, J., Chung, J., & Sohn, K. (2020). Reinforcement Learning for Joint Control of Traffic Signals in a Transportation Network. IEEE Transactions on Vehicular Technology. https://doi.org/10.1109/TVT.2019.2962514

[15] Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., & Wu, D. O. (2020). Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks. IEEE Transactions on Vehicular Technology. https://doi.org/10.1109/TVT.2020.2997896

[16] Ge, H., Gao, D., Sun, L., Hou, Y., Yu, C., Wang, Y., & Tan, G. (2021). Multi-Agent Transfer Reinforcement Learning with Multi-View Encoder for Adaptive Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/tits.2021.3115240

[17] Li, W., & Ni, S. (2022). Train timetabling with the general learning environment and multi-agent deep reinforcement learning. Transportation Research Part B: Methodological. https://doi.org/10.1016/j.trb.2022.02.006

[18] Bretas, A. M. C., Mendes, A., Chalup, S., Jackson, M., Clement, R., & Sanhueza, C. (2023). Addressing deadlock in large-scale, complex rail networks via multi-agent deep reinforcement learning. Expert Systems. https://doi.org/10.1111/exsy.13315

[19] Bretas, A., Mendes, A., Sawah, S., & Chalup, S. (2019). Modelling railway traffic management through multi-agent systems and reinforcement learning. El Sawah, S. (ed.) MODSIM2019, 23rd International Congress on Modelling and Simulation. https://doi.org/10.36334/modsim.2019.b6.bretas

[20] Ning, X., Dong, W., Ji, Y., & Sun, X. (2024). Multi-Objective Multi-Agent Reinforcement Learning for Energy-Aimed Train Scheduling for Urban Rail Transit Network. 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN). https://doi.org/10.1109/INDIN58382.2024.10774535

[21] Wen, L., Hu, L., Zhou, W., Ren, G., & Zhang, N. (2025). Soft Actor-Critic Deep Reinforcement Learning for Train Timetable Collaborative Optimization of Large-Scale Urban Rail Transit Network Under Dynamic Demand. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2025.3525538

[22] Feng, F., Zou, Z., Liu, C., Zhou, Q., & Liu, C. (2023). Forecast of Short-Term Passenger Flow in Multi-Level Rail Transit Network Based on a Multi-Task Learning Model. Sustainability. https://doi.org/10.3390/su15043296

[23] Wu, J., He, D., Li, X., He, S., Li, Q., & Ren, C. (2023). A Time Series Decomposition and Reinforcement Learning Ensemble Method for Short-Term Passenger Flow Prediction in Urban Rail Transit. Urban Rail Transit. https://doi.org/10.1007/s40864-023-00205-1

[24] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Far, B., Abad, Z. S. H., & Bazzan, A. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. Expert Syst. Appl. https://doi.org/10.31224/osf.io/ewxrj

[25] Ibrahim, A. M., Yau, K., Chong, Y.-W., & Wu, C. (2021). Applications of Multi-Agent Deep Reinforcement Learning: Models and Algorithms. Applied Sciences. https://doi.org/10.3390/app112210870

[26] Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2019.2901791

[27] Yang, S. (2023). Hierarchical graph multi-agent reinforcement learning for traffic signal control. Inf. Sci.. https://doi.org/10.1016/j.ins.2023.03.087

[28] Yang, S., Yang, B., Zeng, Z., & Kang, Z. (2023). Causal inference multi-agent reinforcement learning for traffic signal control. Inf. Fusion. https://doi.org/10.1016/j.inffus.2023.02.009

[29] Devailly, F.-X., Larocque, D., & Charlin, L. (2020). IG-RL: Inductive Graph Reinforcement Learning for Massive-Scale Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2021.3070835

[30] Abdoos, M., & Bazzan, A. (2021). Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. Expert Syst. Appl. https://doi.org/10.1016/j.eswa.2021.114580

[31] Bokade, R., Jin, X., & Amato, C. (2023). Multi-Agent Reinforcement Learning Based on Representational Communication for Large-Scale Traffic Signal Control. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3275883

[32] Zhang, Y., Li, S., Yuan, Y., & Yang, L. (2025). Multi-step looks ahead deep reinforcement learning approach for automatic train regulation of urban rail transit lines with energy-saving. Eng. Appl. Artif. Intell. https://doi.org/10.1016/j.engappai.2025.110181

[33] Fang, J., You, Y., Xu, M., Wang, J., & Cai, S. (2023). Multi-Objective Traffic Signal Control Using Network-Wide Agent Coordinated Reinforcement Learning. Expert Syst. Appl. https://doi.org/10.2139/ssrn.4374888

[34] Benhamza, K., Seridi, H., Agguini, M., & Bentagine, A. (2024). A multi-agent reinforcement learning based approach for intelligent traffic signal control. Evol. Syst. https://doi.org/10.1007/s12530-024-09622-4

[35] Hou, L., Huang, D., Cao, J., & Ma, J. (2023). Multi-agent deep reinforcement learning with traffic flow for traffic signal control. Journal of Control and Decision. https://doi.org/10.1080/23307706.2023.2195408

[36] Li, X., Li, J., & Shi, H. (2023). A multi-agent reinforcement learning method with curriculum transfer for large-scale dynamic traffic signal control. Applied Intelligence. https://doi.org/10.1007/s10489-023-04652-y

[37] Zhang, Y., Deekshith, U., Wang, J., & Boedecker, J. (2024). Improving the Efficiency and Efficacy of Multi-Agent Reinforcement Learning on Complex Railway Networks with a Local-Critic Approach. https://doi.org/10.1609/icaps.v34i1.31533

[38] Khozam, S., & Farhi, N. (2023). Deep Reinforcement Q-Learning for Intelligent Traffic Control in Mass Transit. Sustainability. https://doi.org/10.3390/su151411051

[39] Wang, J., & Sun, L. (2023). Multi-objective multi-agent deep reinforcement learning to reduce bus bunching for multiline services with a shared corridor. Transportation Research Part C: Emerging Technologies. https://doi.org/10.1016/j.trc.2023.104309

[40] Zhang, H., Lu, G., Lei, Y., Zhang, G., & Niyitanga, I. (2022). A hybrid framework for synchronized passenger and train traffic simulation in an urban rail transit network. International Journal of Rail Transportation. https://doi.org/10.1080/23248378.2022.2109522

[41] M. Abdur Rahim, J.H. Park. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, 2025. DOI: 10.1177/10775463251320258.

[42] A. Bouzeriba, A. Boulkroune, S. Bououden. Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities. Mathematical Problems in Engineering, 2017. DOI: 10.1155/2017/8045803.

[43] L. Zouari, A. Boulkroune, A. Ibeas. Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. International Journal of Adaptive Control and Signal Processing, 2013. Scopus ID: 84873265173

[44] L. Zouari, A. Ibeas, A. Boulkroune, J. Cao. Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. 2013 5th International Conference on Modelling, Identification and Control (ICMIC), 2013. DOI: 10.1109/ICMIC.2013.6564134.

[45] G. Rigatos, P. Siano, P. Wira, J.D.M. Cecati. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 2023. DOI: 10.1016/j.rico.2023.100028.

[46] L. Zouari, A. Ibeas, A. Boulkroune. Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. 2013 International Conference on Control, Decision and Information Technologies (CoDIT), 2013. DOI: 10.1109/CoDIT.2013.6689656.