

Article

Energy-Efficient Train Control Based on Energy Consumption Estimation Model and Deep Reinforcement Learning

Jia Liu ^{1,†}, Yuemiao Wang ^{1,†}, Yirong Liu ², Xiaoyu Li ², Fuwang Chen ^{2,*} and Shaofeng Lu ^{2,*} ¹ PCI Technology Group Co., Ltd., Guangzhou 510665, China² Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510640, China

* Correspondence: 202221059729@mail.scut.edu.cn (F.C.); lushaofeng@scut.edu.cn (S.L.)

† These authors contributed equally to this work.

Abstract

Energy-efficient Train Control (EETC) strategy needs to meet safety, punctuality, and energy-saving requirements during train operation, and puts forward higher requirements for online use and adaptive ability. In order to meet the above requirements and reduce the dependence on an accurate mathematical model of train operation, this paper proposes a train-speed trajectory-optimization method combining data-driven energy consumption estimation and deep reinforcement learning. First of all, using real subway operation data, the key unit basic resistance coefficient in train operation is analyzed by regression. Then, based on the identified model, the energy consumption experiment data of train operation is generated, into which Gaussian noise is introduced to simulate real-world sensor measurement errors and environmental uncertainties. The energy consumption estimation model based on a Backpropagation (BP) neural network is constructed and trained. Finally, the energy consumption estimation model serves as a component within the Deep Deterministic Policy Gradient (DDPG) algorithm environment, and the action adjustment mechanism and reward are designed by integrating the expert experience to complete the optimization training of the strategy network. Experimental results demonstrate that the proposed method reduces energy consumption by approximately 4.4% compared to actual manual operation data. Furthermore, it achieves a solution deviation of less than 0.3% compared to the theoretical optimal baseline (Dynamic Programming), proving its ability to approximate global optimality. In addition, the proposed algorithm can adapt to the changes in train mass, initial set running time, and halfway running time while ensuring convergence performance and trajectory energy saving during online use.



Academic Editors: Xiaoyuan Zheng, Jiaping Xiao, Lu Bai and Heling Yuan

Received: 17 November 2025

Revised: 12 December 2025

Accepted: 12 December 2025

Published: 16 December 2025

Citation: Liu, J.; Wang, Y.; Liu Y.; Li, X.; Chen, F.; Lu, S. Energy-Efficient Train Control Based on Energy Consumption Estimation Model and Deep Reinforcement Learning.

Electronics **2025**, *14*, 4939.<https://doi.org/10.3390/electronics14244939>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: energy-efficient train control (EETC); Back Propagation (BP); energy consumption estimation model; Deep Deterministic Policy Gradient (DDPG)

1. Introduction

With the continuous acceleration of China's industrialization and urbanization process, China has become a major carbon emitter in the world, and will be under great pressure to save energy and reduce emissions for a long time in the future [1]. In 2019, the transport sector was the second largest source of carbon emissions in China, accounting for 12.42% of the country's total carbon dioxide emissions [2]. In 2023, the traction energy consumption of China's urban rail system reached 12.934 billion kWh, an increase of 1.619 billion kWh compared to 2022, accounting for 51.78% of the total electricity consumption of urban rail

systems. Energy-efficient train control (EETC) optimizes train trajectory and minimizes traction energy in a given schedule through a reasonable combination of acceleration and deceleration, which is of great significance for saving energy in urban rail systems [3]. Therefore, EETC has gradually attracted extensive attention from many scholars. The optimized speed trajectory according to different optimization methods is helpful to reduce energy consumption, improve punctuality of the train, and improve passenger comfort [4,5].

The EETC problem has been studied for decades, and in the early stages of research, scholars used the Pontryagin maximum Principle (PMP) to solve the problem [6–8]. Based on the PMP method, the scholars indirectly derived the four most energy-efficient combination stages of train operation: maximum acceleration, cruising, coasting, and maximum braking, and then iteratively calculated the optimal combination of switching points by numerical calculation method. However, this method has a high computational cost, and it is not suitable when other complex models, such as motor characteristics, are considered. With the continuous progress of hardware facilities and optimization methods, scholars have applied heuristic algorithms such as genetic algorithm [9], simulated annealing algorithm [10], ant colony algorithm [11] and particle swarm algorithm [12] to the EETC problem to obtain train-operation trajectories with better energy saving through computer iterative calculation. These heuristic algorithms do not rely on the strict four-stage structure. Unlike the PMP method, they are not confined to fixed switching points. Instead, they continuously explore the solution space through iterative calculations, resulting in high-quality train-operation trajectories. However, the calculation cost of this method is very high, and it may converge to the local optimal solution during program running, and it is impossible to guarantee that the train-operation trajectories of the same quality can be obtained every time the program runs. With the upgrading of computers, scholars have applied mathematical programming methods such as mixed integer linear programming [13], pseudo-spectrum method [14], and convex optimization [15] to the EETC problem. These methods use piecewise linear or relaxation techniques to simplify the model complexity, and cooperate with specific solvers to solve the optimal train trajectory. However, these methods still have a strong trade-off between model accuracy and solution speed, making it challenging to simultaneously accommodate more complex and precise models while ensuring fast computation. The dynamic programming method, as a classical mathematical programming method, also has a very important application in solving EETC problems [16]. Lu et al. constructed the multi-stage decision sequence and train-running state network of the EETC problem using the traditional dynamic programming method [17]. In each decision stage, the solution with the worst quality is discarded to reduce the search space, so as to obtain the optimal train trajectory faster. This method starts from the starting position of the train to solve the optimal trajectory forward, and the traversal solution space is still huge, so it is difficult to obtain the train-speed trajectory in real time. The dynamic programming method [18] of constructing a lookup table evaluates the status points of the entire train-running space in the offline calculation process and stores them as searchable lookup tables, and then obtains the optimal speed trajectory by searching the tables during online use. Although this method can achieve the effect of offline calculation and online application, it has great limitations on the dimensions of the train-running state. With the increase in the complexity and dimension of the model, the calculation cost increases exponentially.

With the development of computer science, reinforcement learning methods of artificial intelligence have been fully applied to EETC problems [19–23]. Yin et al. proposed an intelligent train driving algorithm based on Q-learning to optimize the train trajectory online, and its optimized trajectory reduced energy consumption by about 10% compared with manual driving trajectories [24]. Su et al. described the EETC problem as a finite

Markov decision process and proposed a strategy optimization method based on soft actor-critic. The deep reinforcement learning algorithm was used to solve the driving strategy, which was superior to Q-learning and Deep Q-learning methods in energy saving and punctuality [25]. Ning et al. proposed a high-speed train trajectory-optimization method based on Deep Deterministic Policy Gradient (DDPG) [26]. Through offline training of the mapping relationship between learning state and action, the experiment showed that the optimization strategy could generate an on-time and energy-saving trajectory under different running times, the calculation time met the real-time requirements, and the energy consumption was significantly lower than that of the optimization algorithm based on expert rules. Recent studies from 2020 to 2025 have further advanced RL applications in EETC [26–31]. For instance, Ning et al. applied DDPG to high-speed trains to handle continuous action spaces [26], while Su et al. utilized Soft Actor-Critic (SAC) to enhance robustness against disturbances [27]. Other works have explored advanced optimization techniques for multi-train regulation [32] or timetable optimization [33]. These methods show that reinforcement learning can meet the requirements of energy saving, punctuality, and online solution in the application of the EETC problem. The reinforcement learning algorithm generates a large amount of data to learn and make decisions [34]. As a part of environmental reward, energy consumption is crucial. However, the existing reinforcement learning research mainly deduces traction energy consumption during train operation based on deterministic mathematical models and uses these model parameters in the optimization process. In real-world metro operations, the traction environment is dynamic and complex. Key physical parameters, particularly the basic resistance coefficients, are subject to temporal drift. For instance, mechanical wear between wheels and rails over long-term operation alters friction characteristics. Furthermore, seasonal or daily variations in ambient temperature and humidity within tunnels can affect air density and rail adhesion, leading to fluctuations in aerodynamic drag and rolling resistance. Relying on a fixed mathematical model fails to capture these time-varying disturbances, resulting in a mismatch between the optimization model and the actual physical environment. As a result, the optimization process of train operation is still highly dependent on the accurate mathematical model parameters of train traction [35,36].

The accuracy and authenticity of model establishment during train operation are the basis of train-operation control, and it is very important for the accuracy of energy consumption calculation and the energy-saving of optimization algorithm in the optimization of EETC problem [37,38]. Wang et al. calibrated their model using train-operation data from Portland, Oregon, and the proposed modeling approach was capable of capturing the energy consumption differences related to train, line, and operational characteristics [39]. The parameters of the mathematical model of train operation directly affect the accuracy of the model. As a key parameter, the basic resistance coefficient of trains has a great impact on the calculation accuracy of train-operation energy consumption. Liu et al. proposed to use the multi-innovative least squares algorithm to identify the basic resistance parameters of subway operation in order to improve the accuracy of parameter estimation [40]. Based on experimental data, Radosavljevic applied the least squares method to estimate the traction characteristics of trains on Serbian lines, which can describe the real situation more accurately than other empirical formulas [41]. On the basis of the model parameters, the energy consumption of the train-running process can be modeled and predicted to obtain the energy consumption estimate. However, the parameters of the mathematical model in the train-running process are interrelated with many factors, so it is difficult to accurately express them using a general empirical formula, and the reliability of parameter estimation is difficult to guarantee [39]. Machine learning technology has powerful nonlinear modeling ability, which can consider multiple factors affecting the operation

energy consumption, and obtain a reliable and effective estimation network through the training of a large number of data sets, which is conducive to improving the accuracy of the estimation of the train-operation energy consumption [42,43].

In order to improve the practicability, adaptability, and energy saving in the EETC problem, a train-speed trajectory-optimization method combining a data-driven energy consumption estimation model and deep reinforcement learning is proposed in this paper, and its performance is studied. In this paper, the important basic resistance coefficient is fitted based on real train-operation data from the Guangzhou Metro, which operates entirely within tunnels, and is used as the data source for generating the training data of the energy consumption estimation model. The proposed energy consumption estimation model based on a BP neural network considers many factors such as train mass, gradient, Curve radius, etc., and achieves an accurate estimation of train-operation energy consumption. As a part of the environment of the DDPG algorithm, the estimation model is fed back to the energy consumption part of the train state transition reward. The proposed DDPG algorithm does not use an accurate basic resistance coefficient as expert knowledge, and considers the influence of train quality, running time, and other factors on speed trajectory optimization. Through offline training and online use, the adaptability and energy saving of the algorithm are discussed. The contributions of this paper are summarized as follows:

- The proposed energy consumption estimation model can accurately reflect the influence of basic resistance on running energy consumption, and has good generalization ability to the change in running environment, such as train mass, gradient, curve radius, and running length.
- The proposed method ensures that all the optional actions are feasible, and has good adaptability, which can adapt to the influence of different train masses and different running times while maintaining the optimization effect of energy consumption.
- The proposed energy consumption estimation model is based on data-driven training, and effectively replaces the accurate mathematical model as a part of the reinforcement learning environment without affecting the energy consumption optimization effect of the running trajectory.

The remainder of this paper is organized as follows. Section 2 presents the research methodology, outlining the train dynamics model, the data-driven energy consumption estimation strategy, and the DDPG-based trajectory-optimization framework. Section 3 details the numerical experiments, including parameter identification, model training results, and comparative analysis. Finally, Section 4 concludes the paper and discusses limitations and future research directions.

2. Methodology

To achieve energy-efficient train control while reducing dependence on precise physical parameters, this paper proposes a methodology combining data-driven modeling and deep reinforcement learning. The overall framework consists of three main stages. First, a Parameter Identification stage uses real operation data to regress the unit basic resistance coefficient, which allows for the generation of a realistic synthetic dataset. Second, an Energy Estimation Modeling stage constructs a Back Propagation (BP) neural network trained on the synthetic dataset (with added Gaussian noise) to accurately map train states to energy consumption. Third, a Trajectory-Optimization stage integrates this pre-trained estimation model into a Deep Deterministic Policy Gradient (DDPG) environment, where the agent learns to optimize the speed trajectory for energy saving and punctuality. The following subsections detail the mathematical models and algorithms used in these stages.

2.1. Assumptions and Problem Conditions

To formulate the EETC problem mathematically and ensure the feasibility of the data-driven approach, the following assumptions and conditions are established:

- **Point Mass Model:** The train is modeled as a single point mass, ignoring the internal coupler forces between carriages. This is a standard assumption in longitudinal train dynamics control.
- **Discretized Environment:** The line conditions, including gradient, curvature, and speed limits, are assumed to be constant within each discretized distance segment.
- **Stable Tunnel Environment:** The study focuses on urban metro systems operating primarily in tunnels (specifically based on Guangzhou Metro data). Therefore, external environmental factors such as strong crosswinds, rain, or snow are considered negligible, creating a relatively stable operating environment compared to open-air railways.

2.2. Train Dynamics Model

The longitudinal dynamic model of a single train can be expressed as:

$$\frac{dv}{dt} = \frac{f_M - M \cdot g(w_D + w_G + w_S + w_L)}{M} \quad (1)$$

where v is the speed of the train, M is the total mass of the train, g is the acceleration due to gravity, and f_M is the traction or braking force applied by the motor. The formula for f_M is as follows:

$$f_M = p_1 \cdot f_t^{max} - p_2 \cdot f_b^{max} \quad (2)$$

where p_1 and p_2 are coefficients from 0 to 1, and their product is 0. It is worth noting that the motor has a complex inherent characteristic curve, and the maximum traction/braking force f_t^{max} and f_b^{max} change with the speed of the train [44]. Additionally, the empirical formula of the unit resistance w_D [45], w_G , w_S , w_L [46] of the train is as follows:

$$w_D = k_A + k_B \cdot v + k_C \cdot v^2 \quad (3)$$

$$w_G = \sin(k_G) \quad (4)$$

$$w_S = \frac{k_S}{R} \quad (5)$$

$$w_L = k_L \cdot L_s \cdot v^2 \quad (6)$$

among them, k_A , k_B and k_C are the coefficient of the unit basic resistance w_D , which mainly comes from air resistance, friction between bearings, friction between train wheels and tracks and shock vibration. w_G represents the additional resistance of the ramp where the train is located, and k_G represents the angle between the ramp where the train is currently located and the horizontal line. Because when the train runs on the curve, due to the action of centrifugal force, the wheel and the track will generate additional friction, and its unit resistance is w_S with the coefficient k_S . In addition, when the train runs in a long L_s tunnel, the pressure difference formed by the head and tail will hinder the train's progress, resulting in additional tunnel resistance, whose unit resistance is w_L with the coefficient k_L [46]. The energy conversion relationship between traction/braking is as follows:

$$\frac{de}{dx} = p_1 \cdot f_t^{max} / \eta_t - p_2 \cdot f_b^{max} \cdot \eta_b \quad (7)$$

where e refers to the energy consumed by the train during its operation; x refers to the line distance; η_t and η_b refer to the motor traction/braking efficiencies. The optimization principle of the EETC problem is to minimize the energy consumption of the route by

controlling the acceleration and deceleration of the train or the motor output on the premise that the train arrives at the terminal time unchanged.

2.3. Unit Basic Resistance Coefficient Fitting

The basic resistance of train operation has a particularly significant impact on energy consumption. This section conducts a regression analysis on the unit basic resistance coefficient based on the real running lines and calculates the optimal value of each coefficient in the quadratic function.

Based on the known gradient, train trajectory, curve radius, and motor force, the unit basic resistance corresponding to different speeds during train operation can be derived backwards from Formula (1), as shown below:

$$w_D = \frac{f_M - M \cdot a}{M \cdot g} - w_G - w_S - w_L \quad (8)$$

where a is the current acceleration of the train. Since the unit's basic resistance varies with the operating speed, there will be more sampling points in a certain speed interval. An appropriate upper limit of sampling number is set in each ΔV km/h speed interval, and polynomial fitting is carried out to finally obtain the unit basic resistance coefficient k_A, k_B, k_C based on the backward regression data. The polynomial is optimized based on the gradient descent method, and the mean square error is used as the loss function of regression analysis:

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^n (w_D^i - \hat{w}_D^i)^2 \quad (9)$$

where n is the number of samples, w_D^i is the unit basic resistance obtained by backward inference based on real data, and $\hat{w}_D^i$ is the estimate of the unit basic resistance by the polynomial.

2.4. Train-Operation Energy Consumption Estimation Model

As shown in (7), the energy consumption of train operation can be calculated after the parameter information of train operation is accurately obtained. However, in the real environment, there are many factors affecting the energy consumption of train operation, and the above equation and corresponding coefficient may not be able to accurately describe the process of train energy consumption. This section proposes a train traction energy consumption estimation model based on a BP neural network, and establishes the relationship between energy consumption and various influencing factors through the data of operation energy consumption. BP neural networks are particularly suitable for this task due to their simple structure, ease of implementation, and effectiveness in handling nonlinear regression problems. Given the relatively low-dimensional input and single-output nature of this estimation task, BP networks provide an efficient and reliable modeling solution compared to more complex models. Due to the lack of a large amount of accurate actual train-running energy consumption data as data support, this paper uses a small amount of actual running data to identify the unit's basic resistance coefficient. Based on this mathematical model (i.e., the polynomial regression model), a large number of energy consumption data were generated, which served as the training data for the energy consumption estimation model.

As shown in Figure 1, the network structure of the energy consumption estimation model consists of train mass, gradient, curve radius, tunnel length [46], running length, current speed, and next speed. The traction energy consumed by a train running at uniform acceleration from the current speed to the next speed is set as the output of the network

under the conditions of specified gradient, curve radius, and tunnel length. The calculation formula of energy consumption data of training can be obtained based on Formula (7):

$$E_{train} = \int_0^{L_r} (p_1 \cdot f_t^{max} / \eta_t - p_2 \cdot f_b^{max} \cdot \eta_b) dx + \epsilon \quad (10)$$

where L_r is the running length in the train input, and ϵ is the noise of the data, which is a random variable from 0 to 0.1 kWh.

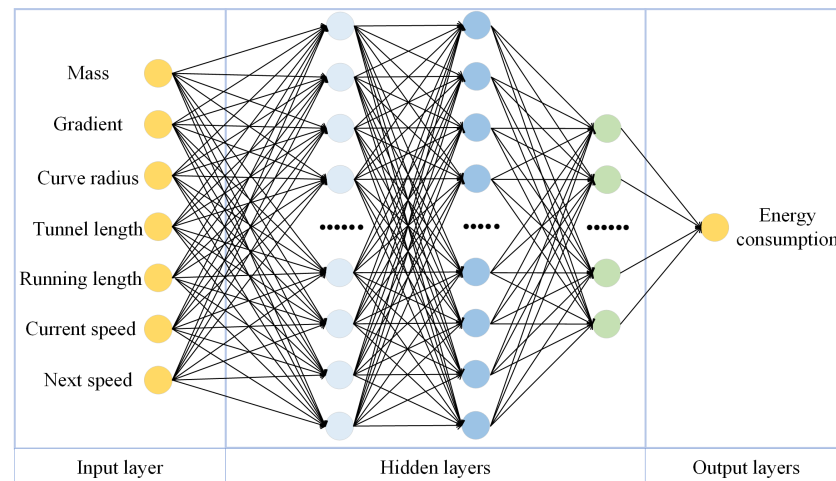


Figure 1. Network structure of energy consumption estimation model.

To ensure the reproducibility of the research, the detailed hyperparameter configuration of the BP neural network is listed in Table 1.

Table 1. Hyperparameter configuration of the BP neural network.

Parameter	Value
Input Layer Nodes	7
Hidden Layers	3
Hidden Layer Nodes	100, 400, 100
Output Layer Nodes	1
Activation Function	Leaky ReLU
Loss Function	MSE
Optimizer	Adam
Learning Rate	0.005
Training Sample Size	1000
Training Iterations	2000

2.5. Design of Reinforcement Learning Algorithm

The DDPG algorithm combines deterministic strategy gradient and deep learning to improve the efficiency of data utilization, integrates the successful experience of the DQN algorithm, uses experience playback and sets a separate target network, making the algorithm widely used in the field of control optimization [47,48]. As shown in Figure 2, it is the control framework of the reinforcement learning algorithm. In order to achieve the purpose of online optimization of train trajectory, this paper studies EETC problem based on DDPG algorithm combined with expert experience of train energy-saving operation and well-trained energy consumption estimation model. The rationale for selecting the DDPG algorithm over discretization-based algorithms (e.g., Deep Q-Network, DQN) lies in the continuous nature of train control. In real-world operations, traction and braking forces are continuous physical quantities. Discretization algorithms require dividing the action space into finite levels, which leads to two major issues:

- **Control Roughness:** Discrete actions can cause abrupt changes in acceleration, resulting in “chattering” control behavior that reduces passenger comfort and increases mechanical wear.
- **Curse of Dimensionality:** To achieve high control precision with DQN, the action space must be finely discretized, leading to an exponential increase in the number of action states and making the model difficult to converge.

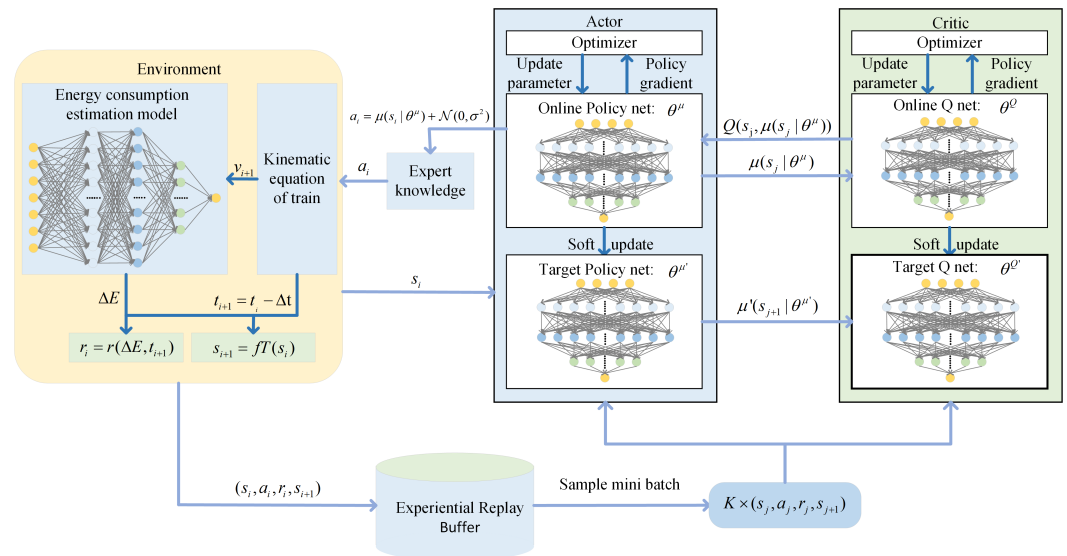


Figure 2. Reinforcement learning framework.

In contrast, DDPG adopts an Actor-Critic architecture that can directly output continuous actions, enabling smooth and precise control of the train’s acceleration while avoiding the dimensionality problem.

2.5.1. Constraint-Based Prior Knowledge and Experience Replay Mechanism

In reinforcement learning algorithms, Constraint-based Prior Knowledge and Experience Replay Mechanism are crucial components that work synergistically to facilitate the agent’s effective learning in complex environments.

By incorporating Constraint-based Prior Knowledge, reliable decision rules and safety constraints are provided, which not only guide the agent in effective exploration but also ensure the stability of the training process. Specifically, the agent possesses the following prior knowledge during train operation:

- **State and Track Information:** The agent can acquire real-time state information, such as position, speed, acceleration, and runtime, as well as track information such as slope and curve radius. This comprehensive environmental awareness allows the agent to make precise decisions based on the real-time state and track conditions.
- **Stopping Requirement:** The agent is required to bring the train’s speed to exactly zero at the endpoint to ensure accurate stopping. This goal sets a clear objective for the agent, guiding it to optimize decisions and ensure stopping precision at the endpoint.
- **Acceleration Safety Constraints:** The agent is fully aware of the maximum and minimum achievable acceleration under given position and speed conditions. This knowledge provides safe boundaries for exploration and optimization, preventing potential risks from exceeding system limits.
- **Time Constraints:** The agent is aware of the minimum remaining running time at each position. If the actual remaining time is below this minimum, the agent will take

appropriate acceleration measures to ensure on-time arrival at the destination. This mechanism helps optimize running efficiency and prevent delays.

These prior knowledge components help the agent make safer, more efficient decisions in complex and dynamic operational environments, ensuring reliable operation in various scenarios.

On the other hand, the Experience Replay Mechanism plays a crucial role in deep reinforcement learning. The DDPG algorithm stores the agent's interactions with the environment in an experience replay buffer and randomly samples these experiences during training to update the policy, thus breaking the temporal and spatial correlation of the data and preventing overfitting. Additionally, DDPG introduces a target network and adjusts its parameters through soft updates, reducing fluctuations during training and accelerating convergence. By combining Constraint-based Prior Knowledge and Experience Replay Mechanism, the reinforcement learning algorithm in this study not only demonstrates higher adaptability and robustness but also ensures stability and convergence during training, effectively optimizing the train's operation.

2.5.2. State Space of the Train

The state space is the basis of the interaction between the train and the environment. Selecting the variables that can accurately describe the state of the train running, as the state of the environment of the train-running can promote the convergence and optimization of the algorithm. The definition of state space selected in this paper is as follows.

$$S = \{s_i = [m, X_i, v_i, t_i] | m \in [\underline{M}, \overline{M}], v_i \in [0, \overline{V}_i], X_i \in [0, X_i], t_i \in [0, T_r]\} \quad (11)$$

where m represents the mass of the train and passengers, and $\underline{M}$ and $\overline{M}$ represent the mass of the train when the train is empty and when the train is full. In order to simplify the problem, this paper divides the running journal into N segments, X_i is the length of the position point i^{th} and the starting point, X_i represents the distance between the starting point and the end point. v_i represents the current speed of the train in position i , which is limited by the upper limit of the speed $\overline{V}_i$ in the current position; t_i is the remaining disposable time of the train at the point of i position, and T_r is the set maximum operating time of the train.

2.5.3. Action Space of the Train

The train mainly regulates the acceleration of the train through the control signal issued by the control handle or the onboard equipment to generate the corresponding traction braking force. The acceleration of the train can be regarded as a continuous variable and becomes a continuous action. The action space selected in this paper is defined as follows:

$$A = \{a_i \in [\underline{A}_i^{v_i}, \overline{A}_i^{v_i}]\} \quad (12)$$

where a_i represents the train's optional action at position i , and $\underline{A}_i^{v_i}$ and $\overline{A}_i^{v_i}$ represent the lower and upper limits of the train's optional action at position i . The values of $\underline{A}_i^{v_i}$ and $\overline{A}_i^{v_i}$ are calculated according to the constraints of motor traction characteristics, maximum speed of the fastest running mode, current train speed, train motion mechanics formula, etc., and stored as a search table to ensure the safety and feasibility of train motion during algorithm exploration. In order for the train to stop accurately at the final position, a_{N-1} is selected for the last stage of the action, whose value must be the acceleration that causes the train to reach the final speed of 0.

In the running process of the algorithm, the online strategy network generates an optimal estimated action value according to the current state of the train and adds the detection noise to form the action of the train. The formula is shown below:

$$a_i = \mu(s_i|\theta^\mu) + \mathcal{N}(0, \sigma^2) \quad (13)$$

Although Ornstein–Uhlenbeck (OU) noise is traditionally used in DDPG to generate temporally correlated exploration; recent studies [49] suggest that uncorrelated Gaussian noise is often sufficient for continuous control tasks. Given the inherent inertia of the train dynamics, which naturally provides temporal correlation in state transitions, we adopted zero-mean Gaussian noise $\mathcal{N}$ for simplicity and stability. In addition, this paper employs a “**Safety Shielding**” (or **Action Masking**) mechanism based on prior expert knowledge. When the running time of the train at the current position is less than the remaining minimum running time, positive random bias is added to the actions to improve the punctuality of train operation, make the detection trajectory more reasonable, and accelerate the algorithm convergence speed.

$$a_i = \begin{cases} a_i & \text{if } t_i > \underline{T}_i, \\ \min(\overline{A}_i^{v_i}, a_i + 0.5 * U(0, \overline{A}_i^{v_i} - \underline{A}_i^{v_i})) & \text{if } t_i \leq \underline{T}_i. \end{cases} \quad (14)$$

where $\underline{T}_i$ is the remaining running time when the train runs at its fastest speed from the starting point to the location i , and $U(0, \overline{A}_i^{v_i} - \underline{A}_i^{v_i})$ is a random value from 0 to $\overline{A}_i^{v_i} - \underline{A}_i^{v_i}$.

2.5.4. Design of Reward Function

Based on the expert experience, train-running time, and the energy consumption generated by the energy consumption estimation model, this paper designs the rewards for train punctuality and energy saving. When the train does not run to the last route segment, the reward design is as follows:

$$r_i = \begin{cases} 1 - \Delta E & \text{if } t_{i+1} > \underline{T}_{i+1}, \\ -\Delta E & \text{if } \underline{T}_{i+1} - 50 \leq t_{i+1} \leq \underline{T}_{i+1}, \\ -0.5 - \Delta E & \text{if } t_{i+1} < \underline{T}_{i+1} - 50. \end{cases} \quad (15)$$

When the train runs to the last route segment, the reward design is as follows:

$$r_{N-1} = -\Delta E - 0.1 \cdot \max(|t_N|, 100) \quad (16)$$

In the design of the reward function, the selection of weight coefficients is critical for balancing the trade-off between energy efficiency and punctuality. To determine the optimal coefficients, we conducted a sensitivity analysis based on trial-and-error experiments:

- **Positive Reward Bias:** Ensuring sufficient positive incentives is crucial for accelerating learning. Our experiments indicated that when the positive reward bias significantly exceeds 1, the algorithm tends to overlook the energy consumption factor (measured in kWh) because the reward magnitude overshadows the energy penalty. Conversely, when the bias is close to 0, the convergence speed becomes slow. A bias value of 1 (and similar magnitudes between 0.1 and 5) was found to ensure stable convergence while maintaining sensitivity to energy costs.
- **Penalty for Abnormal States:** When a significant time deviation occurs, the train is in an abnormal state. A penalty mechanism is needed to constrain the agent. We found

that a penalty coefficient of 0.5 effectively balances the need for constraint without imposing an excessive negative impact that could destabilize the learning process.

- Time Threshold (50 s): The value of 50 represents the critical boundary for time deviation. Prior to training, we calculated the theoretical fastest running time based on expert knowledge. Experimental results showed that setting this threshold to 50 s achieves a good trade-off between exploration capability and stability. If set too large, the agent may over-explore detrimental actions; if set too small, the agent's exploration capability is unduly restricted.

To ensure the reproducibility of the proposed control strategy, the detailed hyperparameter settings and network architecture for the DDPG algorithm are presented. Based on the experimental implementation, both the Actor and Critic networks adopt a deep, fully connected architecture designed to capture the complex mapping between train states and optimal actions. The specific configuration parameters are listed in Table 2.

Table 2. Hyperparameter configuration of the DDPG algorithm.

Parameter	Value
Network Architecture	
Actor Hidden Layers	3
Actor Hidden Nodes	60, 1000, 60
Actor Activation	ReLU (Hidden), Tanh (Output)
Critic Hidden Layers	3
Critic Hidden Nodes	60, 1000, 60
Critic Activation	ReLU (Hidden), Linear (Output)
Training Parameters	
Actor Learning Rate (α_μ)	1×10^{-3}
Critic Learning Rate (α_Q)	1×10^{-3}
Discount Factor (γ)	1.0
Soft Update Parameter (τ)	0.1
Replay Buffer Size	2000
Batch Size	500
Exploration Noise (σ)	0.3

2.5.5. Training and Parameter Updating of Network

As shown in Figure 2, the algorithm interacts with the environment, and the resulting interactive data are stored in the experiential replay buffer. When the strategy network and value network are trained, a certain number of experience data samples are extracted from the experience replay buffer as the training basis. Algorithm 1 shows the implementation process of the algorithm. The actor target network can deduce the value of the next action, and the critic target network can estimate the Q -value function $Q'((s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'}))|\theta^{Q'})$ of the state-action in the next decision stage, and then obtain the target value of the current state value:

$$y_j = r_j + \gamma Q'((s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'}))|\theta^{Q'}) \quad (17)$$

where γ is the discount rate to measure the current value of future rewards. Based on the above data, the online critic network parameters are updated by minimizing the loss function, which is set to:

$$Loss = \frac{1}{K} \sum_j (y_j - Q(s_j, a_j|\theta^Q))^2 \quad (18)$$

where K is the number of samples. The policy gradient of Actor during parameter update can be obtained by combining the Q value function of online Q network:

$$\nabla_{\theta^\mu} J(\theta^\mu) \approx \frac{1}{N} \sum_j \nabla_{\theta^\mu} \mu(s_j | \theta^\mu) \nabla_a Q(s_j, a | \theta^Q) \big|_{a=\mu(s_j)} \quad (19)$$

the parameters of the online network are updated above. For the update of the target network, the soft update mechanism ensures that the parameters can be updated slowly, thus improving the stability of the training and learning process:

$$\theta^{Q'} = \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (20)$$

$$\theta^{\mu'} = \tau \theta^\mu + (1 - \tau) \theta^{\mu'} \quad (21)$$

where τ is the soft update parameter, which is used to adjust the update speed of the target network.

Algorithm 1 DDPG algorithm

- 1: Randomly initialize the Critic network θ^Q and Actor network θ^μ and copy the parameters to the target network $\theta^{Q'}$ and $\theta^{\mu'}$
- 2: **for** episode = 1 **to** Max_episodes **do**
- 3: Initialize the train-running state s_0
- 4: Initializing the action exploration noise $\mathcal{N}$
- 5: **for** $i = 0$ **to** $N - 1$ **do**
- 6: Get action based on current state, policy network, exploration noise:
 $a_i = \mu(s_i | \theta^\mu) + \mathcal{N}(0, \sigma^2)$
- 7: As shown in (14), the current action is adjusted according to the minimum train-running time.
- 8: Generate rewards r_i based on (15) and (16) and calculate the next state s_{i+1}
- 9: Randomly sample K sample data $K \times (s_j, a_j, r_j, s_{j+1})$ from the experiential replay buffer
- 10: Calculate the target value based on the target network:
 $y_j = r_j + \gamma Q'((s_{j+1}, \mu'(s_{j+1} | \theta^{\mu'})) | \theta^{Q'})$
- 11: Minimize the loss function $Loss = \frac{1}{N} \sum_j (y_j - Q(s_j, a_j | \theta^Q))^2$ to update the online critic network
- 12: Update the online policy network according to the sampling gradient:

$$\nabla_{\theta^\mu} J(\theta^\mu) \approx \frac{1}{N} \sum_j \nabla_{\theta^\mu} \mu(s_j | \theta^\mu) \nabla_a Q(s_j, a | \theta^Q) \big|_{a=\mu(s_j)}$$

- 13: Soft update the parameters of the target network:

$$\theta^{Q'} = \tau \theta^Q + (1 - \tau) \theta^{Q'}$$

$$\theta^{\mu'} = \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$$

- 14: **end for**
 - 15: **end for**
-

3. Numerical Experiments

In this section, several experiments are set up to demonstrate the effectiveness of the proposed data-driven energy consumption estimation and deep reinforcement learning methods for solving EETC problems. First, based on the real running data of the Suyuan–Shuixi Line of Guangzhou Metro, the unit basic resistance coefficient of train running is analyzed, which is an important mathematical model parameter of BP neural network training data. On this basis, the proposed BP neural network is trained with the train-running energy consumption data, and the proposed energy consumption estimation model is proven to have a good energy consumption estimation effect and generalization ability. Then, the trained energy consumption estimation model is used as an important

part of the reinforcement learning environment reward to participate in the training of the reinforcement learning algorithm. Compared with the tabular dynamic programming method proposed in the literature [50], the feasibility of the proposed energy consumption estimation model replacing the accurate mathematical model is verified. Finally, online optimization experiments with different train masses and different running times are carried out to verify that the proposed method has good adaptability. The software optimizer runs on a 12th Gen Intel(R) Core(TM) i7-12700F 2.10 GHz processor and RTX 2070 SUPER GPU, python3.8 interpreter on a 16 GB RAM computer.

3.1. Identification of Unit Basic Resistance Coefficient

Figure 3 shows the effect of polynomial regression on the unit basic resistance coefficient. The unit basic resistance formula after regression analysis is $w_D = 1.284 + 0.0366 \cdot v + 0.000177 \cdot v^2$. The goodness of fit is 0.922, the root-mean-square error is 0.303, and the average absolute error is 0.22. The coefficient makes the polynomial have a good fitting effect on the velocity—unit basic resistance data.

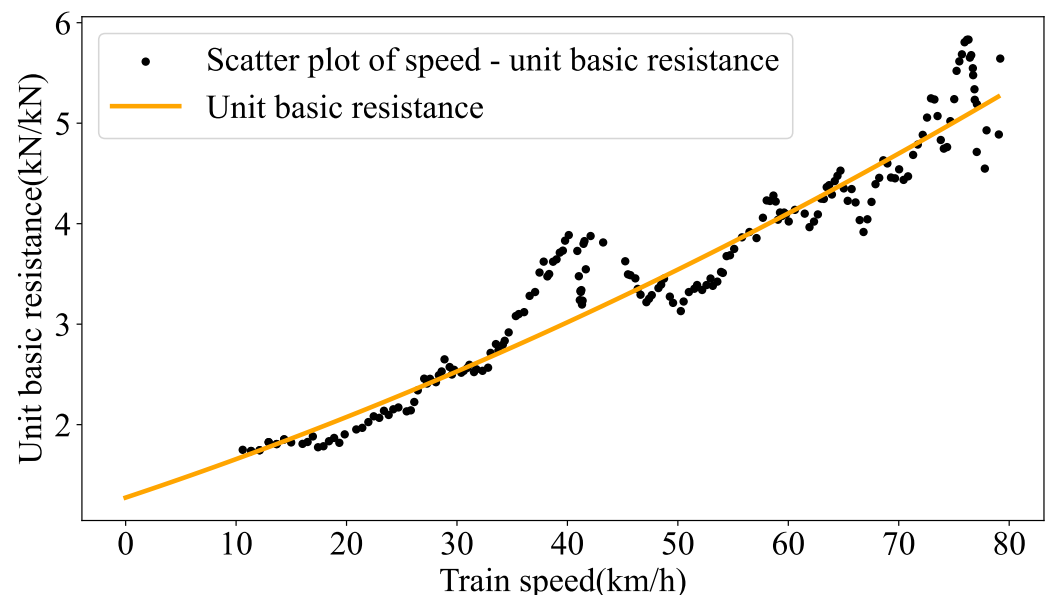


Figure 3. Regression analysis diagram of unit basic resistance coefficient.

As shown in Figure 4, the real running speed trajectory of the train is used as the baseline data, and the tractive force under different conditions is derived backwards based on the train's dynamic equations. To ensure the reliability of the regression analysis, a data preprocessing step was implemented. We analyzed the distribution of the difference between the backward-derived tractive force and the recorded real tractive force. The statistical results indicated that most deviations fell within a reasonable range, while deviations exceeding 25 kN were identified as outliers caused by sensor measurement noise, instantaneous mechanical vibrations, or data synchronization errors. Therefore, we excluded data samples with a traction force difference greater than 25 kN. After this preprocessing, the average deviation of the tractive force without considering basic resistance is 8.75 kN, while the average deviation after considering basic resistance through regression analysis is reduced to 0.11 kN. As shown in Figure 5, the real running tractive force of the train is used as the baseline data, and the speed trajectories under different conditions are derived backwards based on the train's dynamic equations. When the train reaches its final position, the deviation between the backward and actual speed without considering basic resistance is 19.94 km/h, while the deviation after considering basic resistance is reduced

to 0.76 km/h. Using the calibrated unit basic resistance coefficient, the derived tractive force and speed trajectories are compared with the original data. The experimental results show that the calibrated unit's basic resistance coefficient effectively improves the accuracy of the tractive force and speed trajectories, making them closely match the real curves. It is important to note that while the subsequent training data are synthetic, the core physical parameters (resistance coefficients) governing the generation process are derived from real-world operation data. This hybrid approach ensures that the synthetic environment maintains a high degree of fidelity to the actual physical dynamics, thereby bridging the gap between model consistency and reality consistency.

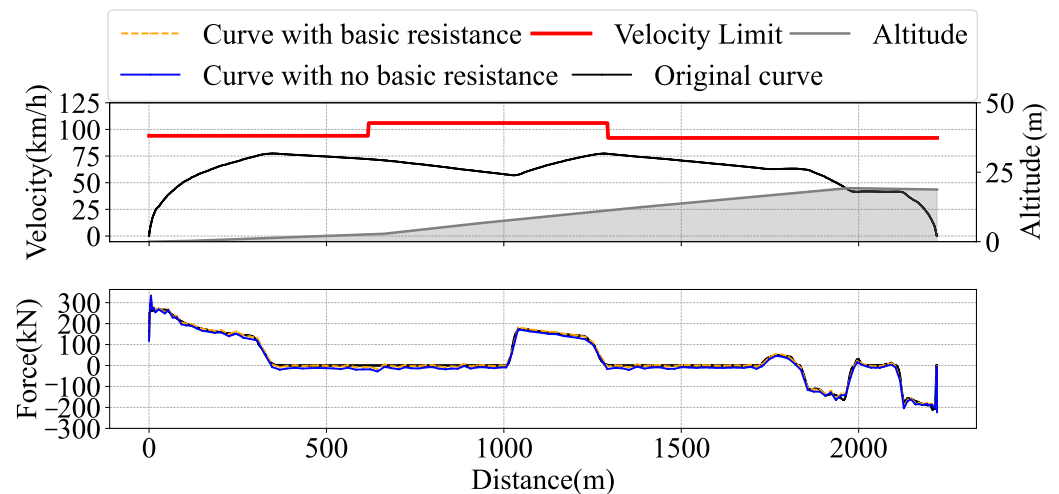


Figure 4. A comparative experiment of train tractive force based on real speed trajectory.

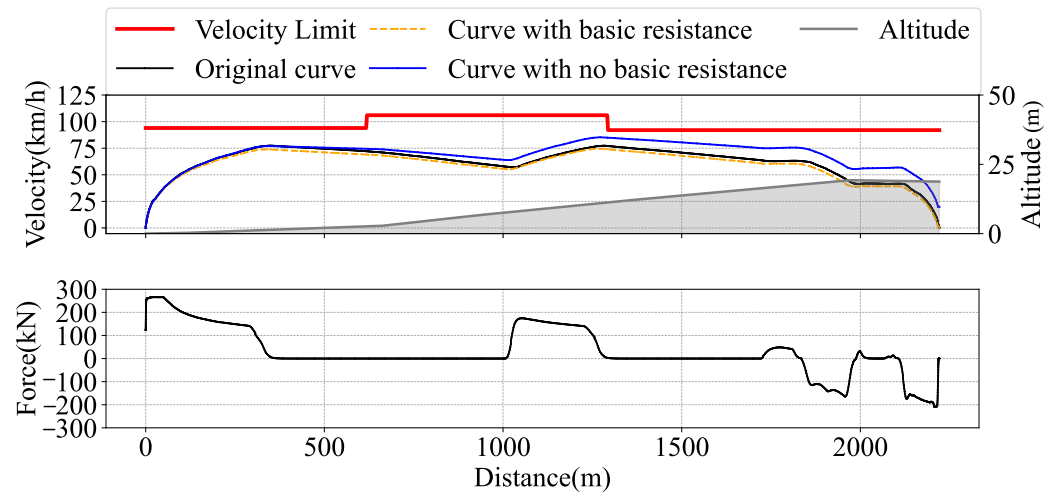


Figure 5. Comparative experiment of train-running speed trajectories derived from real tractive force.

3.2. Data Generation and Preprocessing

To train the BP neural network effectively, a comprehensive synthetic dataset was generated based on the identified resistance model and the train dynamics equation (Equation (10)). The data sampling strategy ensures that the model covers the entire operational state space.

- **Parameter Sampling:** The input state variables were sampled to cover the full operational range of the Guangzhou Metro Line 21. Specifically, the train mass was sampled from [205, 270] t. To ensure the diversity of the training environment, other track parameters were sampled from the specific design specifications of the line: the gradient ranges from $[-30, 30]$ ‰, the curve radius ranges from [300, 8000] m, and the tunnel length

ranges from [1500, 5000] m. Additionally, the running length for energy calculation was sampled within [10, 100] m, and the train speed covered the range of [0, 125] km/h.

- **Noise Injection:** To simulate real-world sensor uncertainties and improve model robustness, Gaussian noise $\mathcal{N}(0, 0.1^2)$ was added to the calculated energy consumption labels.
- **Data Preprocessing:** To accelerate convergence, all input features were normalized to the range [0, 1] using Min-Max scaling before being fed into the network. Outliers in the synthetic generation process (e.g., physically impossible kinematic states) were filtered out.

3.3. Analysis of Training Results of Energy Consumption Estimation Model

As shown in Figure 6, the energy consumption estimation model trains 1 million training data sets each generation, and in the first 100 generations, the average absolute error is stable within 0.1 kWh, which proves that the model has a good convergence effect on the average absolute error of the test data set. After training for 2000 generations, it can be obtained by experiments that the average absolute error of about 1 million test sets is about 0.01 kWh.

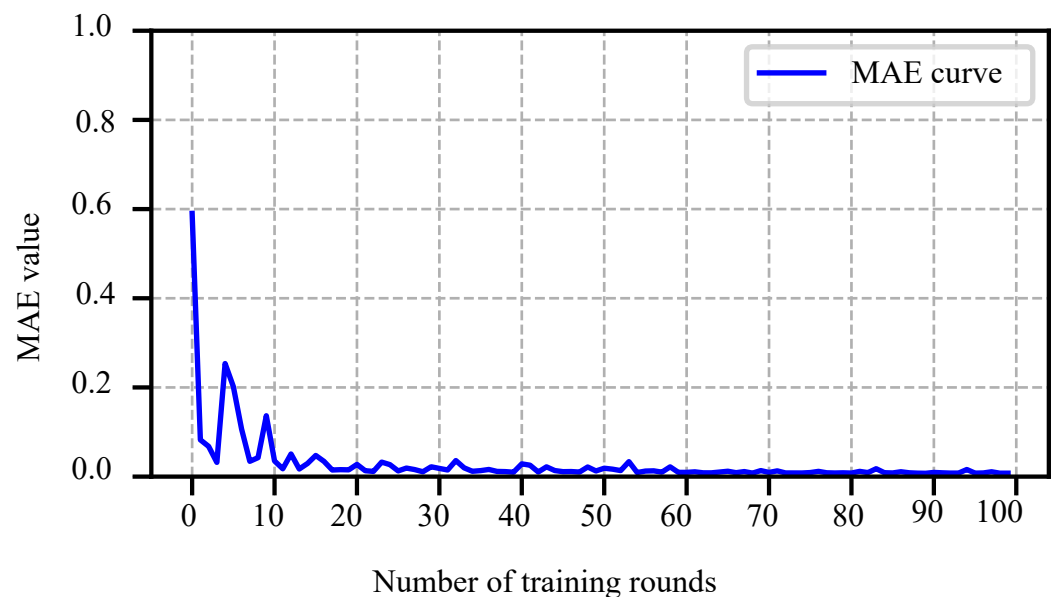


Figure 6. Iteration curve of average absolute error of output energy consumption of the energy consumption estimation model.

As shown in Figure 7, according to the real running track from Tianhe Smart City Station to Jinkeng Station of Guangzhou Metro Line 21, the energy consumed by the train-running model is calculated. As shown in Table 3, for these six journeys, the maximum energy consumption deviation was 0.41 kWh and the minimum energy consumption deviation was 0.16 kWh. The maximum percentage of energy consumption deviation relative to the calculated energy consumption based on the mathematical model is 2.64%, and the minimum is 0.51%. In the six journeys of the train running, the final energy consumption based on the energy consumption estimation model is estimated to be 151.9 kWh, and the final energy consumption based on the mathematical model is estimated to be 150.2 kWh, with an absolute error of 1.6 kWh and a total deviation of about 1.1%. Each estimated step in the operating line is about 60 m, and the number of estimates is about 300 times, with an average deviation of 0.005 kWh in the final energy consumption per estimate. The above results show that the energy consumption estimation model based on real train-speed trajectory is reliable enough.

Table 3. The comparative experiment for estimating the energy consumption of a real trajectory. $\Delta E' = E_{BP} - E_{math}$ and $\delta e = \Delta E' / E_{math} \cdot 100\%$.

	THSC-SZL	SZL-KXC	KXC-SY	SY-SX	SX-CP	CP-JK
E_{math} (kWh)	21.34	15.49	8.40	23.92	32.53	48.56
E_{BP} (kWh)	21.50	15.85	8.62	24.33	32.80	48.81
$\Delta E'$ (kWh)	0.16	0.36	0.22	0.41	0.27	0.25
δe (%)	0.75	2.32	2.64	1.7	0.83	0.51

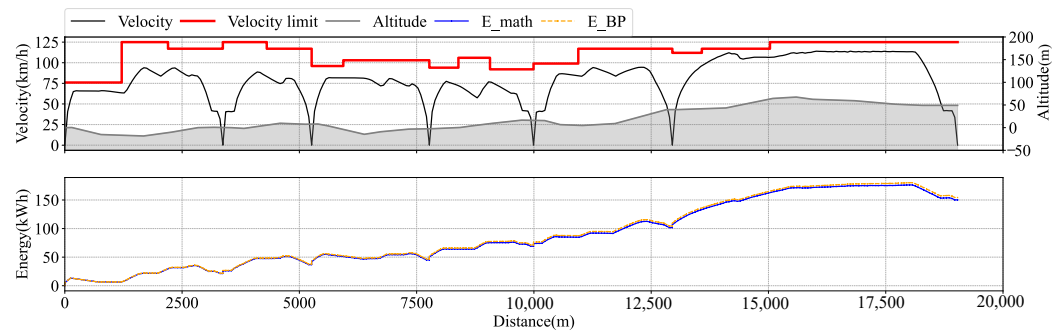


Figure 7. The comparative experiment for estimating the energy consumption of a real trajectory. E_{math} represents the train-operation energy consumption calculated based on the train-operation mathematical model, and E_{BP} represents the train-operation energy consumption derived from the train-operation energy estimation model.

3.4. Reinforcement Learning Trajectory Optimization—Unchanged Initial State

As observed in Figure 8 (and similarly in Figure 9), there are obvious fluctuations in the Loss and Return curves during the early stage. These fluctuations are primarily attributed to the **random exploration noise** added to the actions. This mechanism is essential to prevent the agent from converging to local optima prematurely, encouraging it to explore the action space extensively. As the training progresses, the policy matures, and the exploration noise gradually decreases. After approximately 300 episodes of exploration, the return improves significantly. By the 1600th episode, the fluctuations diminish, and the curves show a steady trend with upward optimization changes in less than 0.01, indicating that the model has achieved stable convergence.

As shown in Figure 10, the velocity fluctuation of the optimization trajectory of the reinforcement learning algorithm based on the mathematical model and the energy estimation model is stronger than that of the optimization trajectory based on the dynamic programming method, but the three are similar in the trend of the force trajectory. The energy consumption of velocity trajectory optimization based on a mathematical model is 23.30 kWh, that of velocity trajectory optimization based on an energy estimation model is 23.32 kWh, and that of dynamic programming is 23.26 kWh. The energy consumption deviation of the three is within 0.1 kWh, which is almost the same. However, the velocity trajectory of the real curve is quite different from that of other tracks, and the energy consumption is 24.4 kWh, which consumes more than 1 kWh of extra energy. In terms of energy saving, the effect of the optimized speed trajectories is similar and far better than the real speed trajectories. To further verify the feasibility of the proposed method for online application, we recorded the computational time required for a single inference step. Based on the experimental hardware environment (Intel i7-12700F CPU and RTX 2070 SUPER GPU), the average computation time for generating a single control action is approximately **0.002 s (2 ms)**. In practical engineering, the control cycle of an Automatic Train-Operation (ATO) system is typically set between **100 ms and 500 ms** to ensure precise tracking and passenger comfort. The inference time of our model is significantly lower than the minimal real-time control requirement ($2 \text{ ms} \ll 100 \text{ ms}$), leaving sufficient margin for

data communication and safety checks. This empirical evidence confirms that the proposed data-driven control strategy is computationally efficient enough to be deployed for real-time online train control.

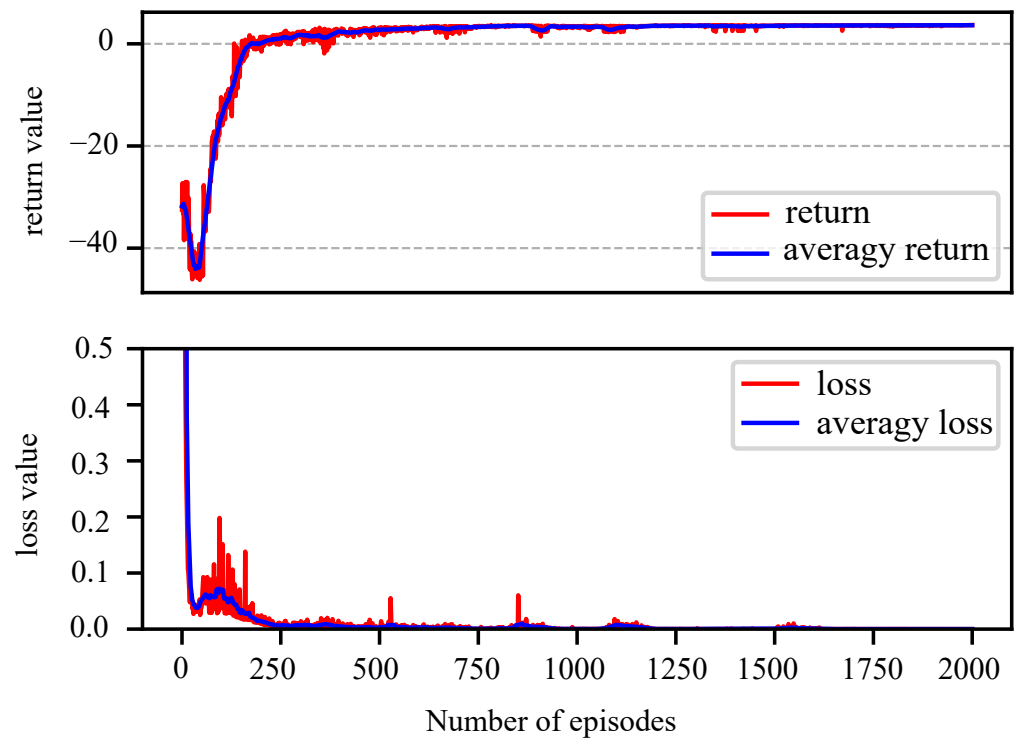


Figure 8. The return and loss function curves of the reinforcement learning algorithm whose initial state does not change during training.

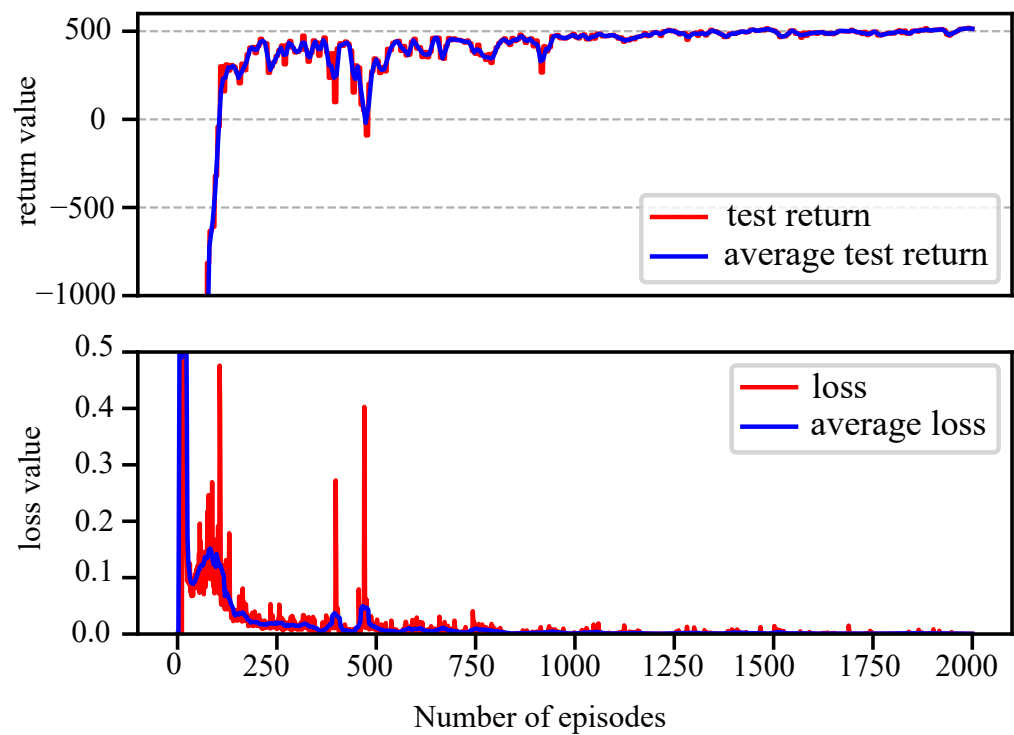


Figure 9. The return and loss function curves of reinforcement learning algorithm whose initial state changes during training. The initial state of the train is a combination of mass intervals of 5 t from 205 t to 270 t and time intervals of 10 s from 130 s to 200 s, and the test return is the total return of 112 initial state optimization tracks.

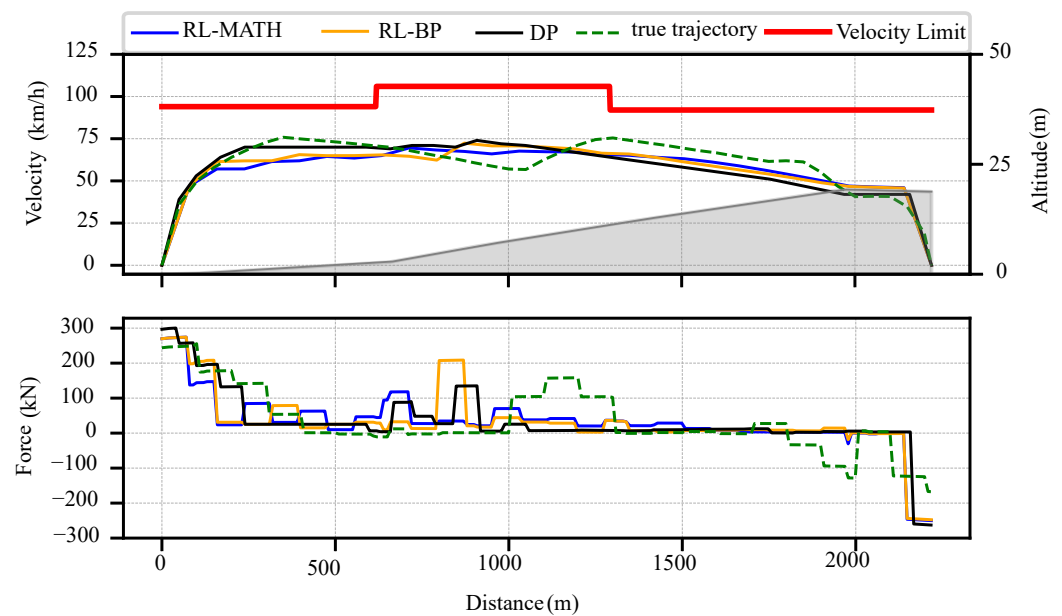


Figure 10. The comparison experiment of reinforcement learning algorithm optimization trajectory, dynamic programming algorithm optimization trajectory and real running trajectory. The energy consumption is calculated according to the velocity trajectory based on the mathematical model.

3.5. Reinforcement Learning Trajectory Optimization—Changed Initial State

As shown in Figure 9, at 150 episodes, after continuous training of the strategy network, the test returns are in a state of rapid growth. Since the algorithm needs to adapt to changes in initial running time and initial train mass, the problem optimized by the agent is more difficult, and the volatility of test returns is strong before 1000 episodes. After 1000 episodes, the loss function gradually converges steadily, and the upward optimization ability is greatly weakened. As shown in Figure 11, the energy consumption assessment model based on a BP neural network is used as the DDPG algorithm environment, and the initial state of change is designed during the training process. The energy-saving performance of the optimized speed trajectory with different initial running time and initial mass is roughly between that of the dynamic programming algorithm with a unit speed of 0.1 m/s and that of the dynamic programming algorithm with a unit speed of 1 m/s. In addition, the algorithm has good adaptability to different train masses, and the optimization effect above 220 t is almost equivalent to that of the dynamic programming algorithm with a unit speed of 0.1 m/s.

As shown in Figure 12, after the train receives the signal of early or late arrival at the terminal at 800 m, the algorithm can adjust the running state in real time and generate the optimized speed trajectory of the subsequent sections. A curve that increases the planned time midway and decelerates appropriately at nodes to delay arrival at the end. A curve that reduces the planned time halfway, however, accelerates appropriately at the node to reach the end ahead of schedule. As shown in Table 4, compared to the original trajectory, the optimized trajectory can reduce energy by about 0.5 kWh through braking recovery after increasing the setting time by 20 s. In addition, the deviation of the end time of the adjusted trajectory is about 1 s. The trajectory still maintains a certain optimization effect of energy saving and punctuality.

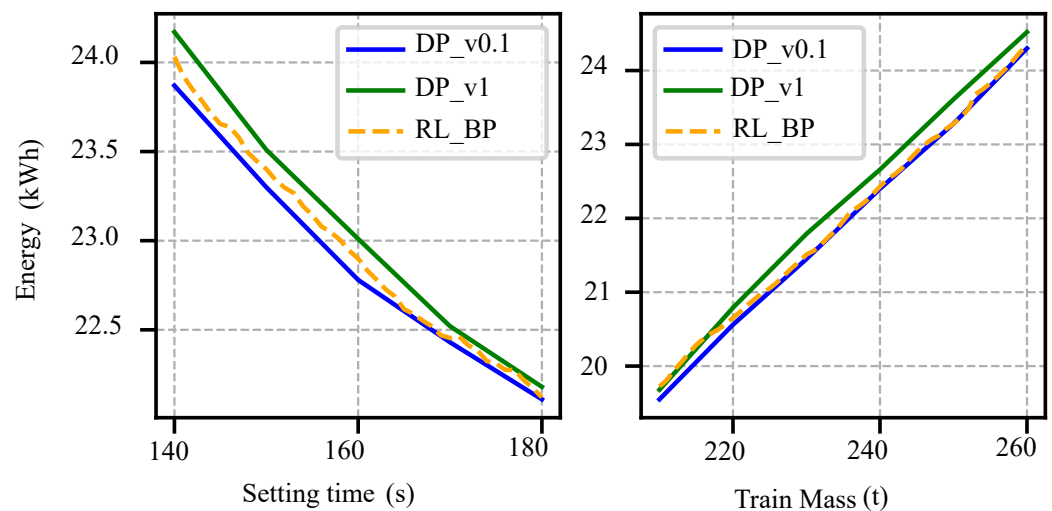


Figure 11. Comparison experiment of energy optimization effect under different running times and train masses. DP_v0.1 means that the discrete unit speed of the train is 0.1 m/s, while DP_v1 means that the unit discrete speed of the train is 1 m/s.

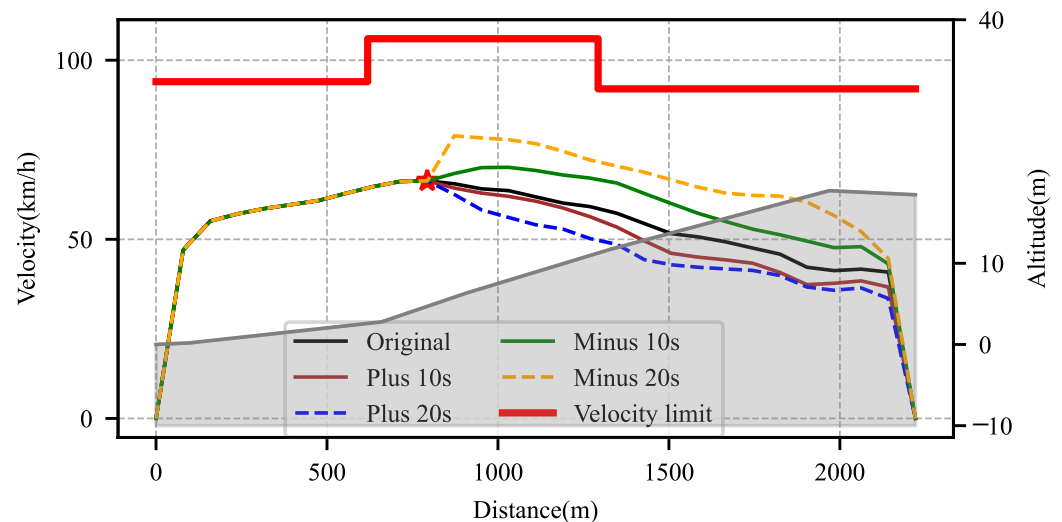


Figure 12. Comparison experiment of real-time adjustment of train trajectory after receiving a delay and an early arrival signal at 800 m. Plus indicates that a late arrival signal is received. Minus indicates that a signal is received that a late arrival signal is received.

Table 4. Experimental results of online adaptation of the algorithm to the change in the running time of the midway plan. The time in the third row represents the absolute deviation between the actual arrival time and the final set arrival time.

E/T	−20 s	−10 s	Original	+10 s	+20 s
Energy (kWh)	24.87	24.00	23.28	22.98	22.76
Time (s)	1.0 s	1.2 s	0.5	0.9	0.3

4. Conclusions

This paper presents a train-speed trajectory-optimization method based on the combination of a data-driven energy consumption estimation model and the DDPG algorithm. The basic resistance in train operation is an important source of energy consumption in train operation. Based on the real subway operation data, this paper analyzes the key unit basic resistance coefficient in train operation. Based on the identified mathematical model, the experimental data of train-operation energy consumption with random noise are

generated, and a reliable and effective BP neural network is obtained through the training of a large number of data sets to complete the data-driven energy consumption estimation model. Finally, based on the DDPG algorithm, combined with expert knowledge and an energy consumption estimation model, the algorithm designed the action adjustment mechanism and reward, completed the optimization training of the reinforcement learning strategy network, and obtained the optimal strategy network. The experiment shows that the identified basic resistance of the train can reflect the real operation of the train to a certain extent, and its goodness of fit is above 0.9, which is suitable for the training source of the energy consumption estimation model. The training process of the energy consumption estimation model has a good convergence effect, which can adapt to the influence of different train masses, gradients, curve radii, tunnel lengths, and running lengths. The average absolute deviation of the test set is about 0.01 kWh, which represents the difference between the energy consumption estimated by the model and that generated by the mathematical model. In terms of the energy consumption estimation of the real speed track data, the total deviation of energy consumption of the six journeys is about 1.1%. The energy consumption deviation of the optimization trajectory of the reinforcement learning algorithm based on the energy consumption estimation model, the reinforcement learning algorithm based on a mathematical model, and the dynamic programming algorithm is less than 0.1 kWh. In addition, the reinforcement learning algorithm based on the energy consumption estimation model can be adaptive to the changes in different train masses, different initial train-running times, and mid-train-running times, and can adapt to the impact of different state changes in real time, while maintaining energy saving and punctuality.

However, there are certain limitations to this study that should be noted:

- **Dependence on Synthetic Data:** Although the training data are synthetic, it is generated based on parameters strictly regressed from real-world operation data of Guangzhou Metro. The simulation environment functions as a “**Digital Twin**” of the specific track section, maintaining high fidelity to the actual physical dynamics. However, we acknowledge that the gap between this high-fidelity simulation and the stochastic raw sensor data in real-time operations still requires further validation.
- **Model Simplicity:** Compared to modern deep learning techniques (e.g., LSTMs or Transformers) that capture long-term temporal dependencies, the proposed BP neural network is relatively simple. While this ensures computational efficiency for real-time control, it may limit the prediction accuracy under highly dynamic transient conditions.
- **Sim-to-Real Gap:** There is a potential risk of the RL policy overfitting to the estimated energy model rather than the true physical system (the “Sim-to-Real” gap). Although our resistance regression mitigates this, further validation on real-world hardware is required.
- **Computational Load:** While our experimental analysis (Section 3.3) shows a low inference time (2 ms), this is based on a PC environment. A rigorous analysis of the computational load on embedded onboard controllers remains to be conducted.

Future work can use a large number of real train-operation data to train energy consumption estimation models to achieve better results. In addition, the influence of power grid, energy storage, and other factors on the EETC problem can be considered. Crucially, regarding the model’s generalization boundary: While the model shows good adaptability to internal state changes (mass and timing), applying it to lines with drastically different characteristics (e.g., different traction systems or open-air environments) may require transfer learning techniques. Exploring transfer learning to enhance the model’s cross-line generalization capability will be a key focus of our future work.

Author Contributions: Conceptualization, writing, editing, and discussion: J.L., Y.W., Y.L., X.L., and F.C.; formal analysis, investigation, and resources, J.L. and Y.W.; data curation, Y.L. and X.L.; methodology, software, and validation: J.L., Y.W., and F.C.; supervision and funding acquisition: S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Research and Development Program of China (No. 2025YFE0105300), National Natural Science Foundation of China (No. 52472343), Guangzhou Municipal Science and Technology Project (No. 2025B03J0017), 2025 Fundamental Research Funds for the Central Universities–Natural Science Category (Grant No. 2025ZYGXZR024) and 2023 Natural Science Foundation of Guangdong Province, China (Grant No. 2023A1515012949).

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Jia Liu and Yuemiao Wang were employed by the company PCI Technology Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

EETC	Energy-efficient Train Control
BP	Back Propagation
DDPG	Deep Deterministic Policy Gradient
PMP	Pontryagin maximum Principle
DQN	Deep Q-Network

References

1. Wang, B.; Sun, Y.; Chen, Q.; Wang, Z. Determinants analysis of carbon dioxide emissions in passenger and freight transportation sectors in China. *Struct. Chang. Econ. Dyn.* **2018**, *47*, 127–132. [\[CrossRef\]](#)
2. Li, J.; Sun, X.; Cong, W.; Miyoshi, C.; Ying, L.C.; Wandelt, S. On the air-HSR mode substitution in China: From the carbon intensity reduction perspective. *Transp. Res. Part A Policy Pract.* **2024**, *180*, 103977. [\[CrossRef\]](#)
3. González-Gil, A.; Palacin, R.; Batty, P.; Powell, J. A systems approach to reduce urban rail energy consumption. *Energy Convers. Manag.* **2014**, *80*, 509–524. [\[CrossRef\]](#)
4. Yin, J.; Tang, T.; Yang, L.; Xun, J.; Huang, Y.; Gao, Z. Research and development of automatic train operation for railway transportation systems: A survey. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 548–572. [\[CrossRef\]](#)
5. Wang, Y.; De Schutter, B.; van den Boom, T.J.; Ning, B. Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach. *Transp. Res. Part C Emerg. Technol.* **2013**, *29*, 97–114. [\[CrossRef\]](#)
6. Ichikawa, K. Application of optimization theory for bounded state variable problems to the operation of train. *Bull. JSME* **1968**, *11*, 857–865. [\[CrossRef\]](#)
7. Howlett, P. An optimal strategy for the control of a train. *ANZIAM J.* **1990**, *31*, 454–471. [\[CrossRef\]](#)
8. Khmelnitsky, E. On an optimal control problem of train operation. *IEEE Trans. Autom. Control* **2000**, *45*, 1257–1266. [\[CrossRef\]](#)
9. He, D.; Guo, S.; Chen, Y.; Liu, B.; Chen, J.; Xiang, W. Energy efficient metro train running time rescheduling model for fully automatic operation lines. *J. Transp. Eng. Part A Syst.* **2021**, *147*, 04021032. [\[CrossRef\]](#)
10. Deng, L.; Cai, L.; Zhang, G.; Tang, S. Energy consumption analysis of urban rail fast and slow train modes based on train running curve optimization. *Energy Rep.* **2024**, *11*, 412–422. [\[CrossRef\]](#)
11. Huang, Y.; Yang, C.; Gong, S. Energy optimization for train operation based on an improved ant colony optimization methodology. *Energies* **2016**, *9*, 626. [\[CrossRef\]](#)
12. Yildiz, A.; Arıkan, O.; Keskin, K. Traction energy optimization considering comfort parameter: A case study in Istanbul metro line. *Electr. Power Syst. Res.* **2023**, *218*, 109196. [\[CrossRef\]](#)
13. Peng, Y.; Lu, S.; Chen, F.; Liu, X.; Tian, Z. Energy-efficient train control incorporating inherent reduced-power and hybrid braking characteristics of railway vehicles. *Transp. Res. Part C Emerg. Technol.* **2024**, *163*, 104626. [\[CrossRef\]](#)
14. Goverde, R.M.; Scheepmaker, G.M.; Wang, P. Pseudospectral optimal train control. *Eur. J. Oper. Res.* **2021**, *292*, 353–375. [\[CrossRef\]](#)

15. Feng, M.; Huang, Y.; Lu, S. Eco-driving strategy optimization for high-speed railways considering dynamic traction system efficiency. *IEEE Trans. Transp. Electrification* **2023**, *10*, 1617–1627. [\[CrossRef\]](#)
16. Haahr, J.T.; Pisinger, D.; Sabbaghian, M. A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points. *Transp. Res. Part B Methodol.* **2017**, *99*, 167–182. [\[CrossRef\]](#)
17. Lu, S.; Hillmans, S.; Ho, T.K.; Roberts, C. Single-train trajectory optimization. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 743–750. [\[CrossRef\]](#)
18. Ghaviha, N.; Bohlin, M.; Holmberg, C.; Dahlquist, E.; Skoglund, R.; Jonasson, D. A driver advisory system with dynamic losses for passenger electric multiple units. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 111–130. [\[CrossRef\]](#)
19. Zhou, K.; Song, S.; Xue, A.; You, K.; Wu, H. Smart train operation algorithms based on expert knowledge and reinforcement learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 716–727. [\[CrossRef\]](#)
20. Zhao, Z.; Xun, J.; Wen, X.; Chen, J. Safe reinforcement learning for single train trajectory optimization via shield SARSA. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 412–428. [\[CrossRef\]](#)
21. Tang, H.; Wang, Y.; Liu, X.; Feng, X. Reinforcement learning approach for optimal control of multiple electric locomotives in a heavy-haul freight train: A Double-Switch-Q-network architecture. *Knowl.-Based Syst.* **2020**, *190*, 105173. [\[CrossRef\]](#)
22. Zhang, H.; Xu, K.; Huang, D.; He, D.; Wu, S.; Xian, G. Hybrid decision-making for intelligent high-speed train operation: A boundary constraint and pre-evaluation reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 17979–17992. [\[CrossRef\]](#)
23. Chen, X.; Guo, X.; Meng, J.; Xu, R.; Li, S.; Li, D. Research on ATO control method for urban rail based on deep reinforcement learning. *IEEE Access* **2023**, *11*, 5919–5928. [\[CrossRef\]](#)
24. Yin, J.; Chen, D.; Li, L. Intelligent train operation algorithms for subway by expert system and reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2561–2571. [\[CrossRef\]](#)
25. Busetto, R.; Lucchini, A.; Formentin, S.; Savaresi, S.M. Data-driven optimal tuning of BLDC motors with safety constraints: A set membership approach. *IEEE/ASME Trans. Mechatronics* **2023**, *28*, 1975–1983. [\[CrossRef\]](#)
26. Ning, L.; Zhou, M.; Hou, Z.; Goverde, R.M.; Wang, F.Y.; Dong, H. Deep Deterministic Policy Gradient for High-Speed Train Trajectory Optimization. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11562–11574. [\[CrossRef\]](#)
27. Zhu, Q.; Su, S.; Tang, T.; Xiao, X. Energy-efficient train control method based on soft actor-critic algorithm. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 2423–2428.
28. Yin, Y.; Wang, Z.; Zheng, L.; Su, Q.; Guo, Y. Autonomous UAV navigation with adaptive control based on deep reinforcement learning. *Electronics* **2024**, *13*, 2432. [\[CrossRef\]](#)
29. Sresakoolchai, J.; Kaewunruen, S. Railway infrastructure maintenance efficiency improvement using deep reinforcement learning integrated with digital twin based on track geometry and component defects. *Sci. Rep.* **2023**, *13*, 2439. [\[CrossRef\]](#)
30. Seo, J.; Kim, S.; Jalalvand, A.; Conlin, R.; Rothstein, A.; Abbate, J.; Erickson, K.; Wai, J.; Shousha, R.; Kolemen, E. Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature* **2024**, *626*, 746–751. [\[CrossRef\]](#)
31. Zhang, C.; Zhang, W.; Wu, Q.; Fan, P.; Fan, Q.; Wang, J.; Letaief, K.B. Distributed deep reinforcement learning based gradient quantization for federated learning enabled vehicle edge computing. *IEEE Internet Things J.* **2024**, *12*, 4899–4913. [\[CrossRef\]](#)
32. Niu, W.; Zhou, Y.; Jiao, X.; Fujita, H.; Aljuaid, H. Trajectory optimization of train cooperative energy-saving operation using a safe deep reinforcement learning approach. *Appl. Intell.* **2025**, *55*, 651. [\[CrossRef\]](#)
33. Li, H.; Yin, J.; Tang, T.; D'Ariano, A.; You, M. Integrated Optimization of Energy-Efficient Timetable and Speed Profiles for Train Platoons in Urban Rail Transit Systems. In Proceedings of the 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), Edmonton, AB, Canada, 24–27 September 2024; pp. 4082–4088.
34. Jia, C.; He, H.; Zhou, J.; Li, J.; Wei, Z.; Li, K.; Li, M. A novel deep reinforcement learning-based predictive energy management for fuel cell buses integrating speed and passenger prediction. *Int. J. Hydrogen Energy* **2025**, *100*, 456–465. [\[CrossRef\]](#)
35. Cunillera, A.; Bešinović, N.; Lentink, R.M.; van Oort, N.; Goverde, R.M. A literature review on train motion model calibration. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3660–3677. [\[CrossRef\]](#)
36. Cunillera, A.; Bešinović, N.; van Oort, N.; Goverde, R.M. Real-time train motion parameter estimation using an unscented Kalman filter. *Transp. Res. Part C Emerg. Technol.* **2022**, *143*, 103794. [\[CrossRef\]](#)
37. Zhao, X.H.; Ke, B.R.; Lian, K.L. Optimization of train speed curve for energy saving using efficient and accurate electric traction models on the mass rapid transit system. *IEEE Trans. Transp. Electrification* **2018**, *4*, 922–935. [\[CrossRef\]](#)
38. Xiao, Z.; Wang, Q.; Sun, P.; You, B.; Feng, X. Modeling and energy-optimal control for high-speed trains. *IEEE Trans. Transp. Electrification* **2020**, *6*, 797–807. [\[CrossRef\]](#)
39. Wang, J.; Rakha, H.A. Electric train energy consumption modeling. *Appl. Energy* **2017**, *193*, 346–355. [\[CrossRef\]](#)
40. Liu, X.; Ning, B.; Xun, J.; Wang, C.; Xiao, X.; Liu, T. Parameter identification of train basic resistance using multi-innovation theory. *IFAC-PapersOnLine* **2018**, *51*, 637–642. [\[CrossRef\]](#)

41. Radosavljevic, A. Measurement of train traction characteristics. *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit* **2006**, *220*, 283–291. [[CrossRef](#)]
42. Fernández, P.M.; Román, C.G.; Franco, R.I. Modelling electric trains energy consumption using Neural Networks. *Transp. Res. Procedia* **2016**, *18*, 59–65. [[CrossRef](#)]
43. Pineda-Jaramillo, J.; Martínez-Fernández, P.; Villalba-Sanchis, I.; Salvador-Zuriaga, P.; Insa-Franco, R. Predicting the traction power of metropolitan railway lines using different machine learning models. *Int. J. Rail Transp.* **2021**, *9*, 461–478. [[CrossRef](#)]
44. Peng, Y.; Chen, F.; Chen, F.; Wu, C.; Wang, Q.; He, Z.; Lu, S. Energy-efficient train control: A comparative study based on permanent magnet synchronous motor and induction motor. *IEEE Trans. Veh. Technol.* **2024**, *73*, 16148–16159. [[CrossRef](#)]
45. Davis, W.J. The tractive resistance of electric locomotives and cars. *Gen. Electr. Rev.* **1926**, *29*, 685–707.
46. Zhou, K. Research on the Car-Ground Simulation System of the LKJ-15 Train Operation Monitoring System Simulation Test Platform. Master's Thesis, Southwest Jiaotong University, Chengdu, China, 2022.
47. Li, K.; Zhou, J.; Jia, C.; Yi, F.; Zhang, C. Energy sources durability energy management for fuel cell hybrid electric bus based on deep reinforcement learning considering future terrain information. *Int. J. Hydrogen Energy* **2024**, *52*, 821–833. [[CrossRef](#)]
48. Jia, C.; Liu, W.; He, H.; Chau, K. Superior energy management for fuel cell vehicles guided by improved DDPG algorithm: Integrating driving intention speed prediction and health-aware control. *Appl. Energy* **2025**, *394*, 126195. [[CrossRef](#)]
49. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
50. Chen, F.; Peng, Y.; Lu, S. Energy-Efficient Train Control Based on Improved Dynamic Programming Algorithm for Online Applications. In Proceedings of the International Conference on Electrical and Information Technologies for Rail Transportation, Beijing, China, 19–21 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 518–529.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.