Application of DDQN Algorithm to Optimization of Train Traction Control in High-Speed Railways

Xikai Liu, Xuelei Meng

***Abstract*—In order to address the issue of enhancing the utilization of traction energy consumption in high-speed railway trains while ensuring safety and punctuality, a methodology is proposed that is founded on the Double Deep Q-Network (DDQN) algorithm. The methodology is initiated with the construction of a high-speed railway train traction control model, followed by the conversion of the entire train operation process into a Markov decision-making process. The train operation simulation environment, state action value function and reward function are then designed according to the fundamental composition of a reinforcement learning algorithm. The subsequent section details the optimization method of high-speed railway train traction control based on DDQN algorithm, trained by combining actual line and train operation data. Finally, a section of a bureau between two stations is selected for example analysis. The results demonstrate that the proposed method achieves a traction energy saving of 11.74kW•h in comparison to the conventional DQN algorithm, while ensuring on-time and stop location accuracy. Furthermore, the study investigates the impact of two hyperparameters, the learning rate α and the selection probability decay value Δε, on the training effect of the model are analyzed, and the results show that with the order of magnitude of the learning rate α of 0.001 and the selection probability decay value Δε of 1/2500, the model has the highest average training reward value and the best training effect.**

***Index Terms*—High-speed railway; Traction control; Energy optimization; Reinforcement learning; DDQN algorithm**

I. Introduction

HIGH-speed railway has the advantages of high-speed and large transport capacity and plays a backbone role in medium and long-distance transport in China. However, there is a concomitant increase in energy consumption because of the operation of these railways. Research indicates that the energy consumption of train traction accounts for approximately 80% of the energy consumption of railway operation [1], emphasizing the significance of reducing train traction energy consumption for the railway industry to minimize expenses and achieve the objective of "double carbon". The primary approach to reducing train traction energy consumption is to optimize the algorithm of the ATO system to provide a more energy-efficient traction control strategy.

Manuscript received November 5, 2024; revised March 13 , 2025.

This work was supported by the Science and Technology Program of Gansu Province (No.: 24JRRA865).

Xikai Liu is a postgraduate student at School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China. (e-mail: m17789388392@163.com).

Xuelei Meng is a Professor at School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China. (e-mail: mxl@mail.lzjtu.cn).

For a single high-speed train running in a zone, the traction control strategy is reflected in the sequence of traction conditions during the whole operation. The method to reduce the traction energy consumption is to output the traction conditions based on a certain state step and find the sequence of conditions with the lowest traction energy consumption under the premise of ensuring that the train arrives at the destination safely and on time.

In 1990, [2] determined through research that the optimal sequence of traction conditions for train operation in a section is maximum power traction—constant speed cruising—inertial coasting—maximum power braking. Subsequent scholars at home and abroad have conducted extensive and in-depth studies on the energy saving operation control of trains, with many research methods based on genetic algorithms, particle swarm optimization algorithms, and heuristic algorithms. A numerical algorithm for train operation control that combines offline approximate dynamic programming (ADP), and online search processes was suggested by [3]. A multi-objective particle swarm optimization algorithm was used in [4] to solve the Pareto front of train energy consumption and travel time, while considering the impact of mechanical efficiency during traction and braking, the effect of contact network resistance on energy consumption, and the influence of train regenerative braking energy on overall energy consumption. In [5], an adaptive improved firefly algorithm for train energy saving driving was proposed with a view to conducting multi-objective optimization of train operation trajectories. In [6], the golden ratio genetic algorithm was utilized to solve the target speed and train condition transition points, with the constraints of electric division being taken into consideration. A high-speed train operation manipulation method based on the differential evolution algorithm was suggested by [7], which achieves multi-objective optimization of energy saving, punctuality, and passenger comfort by solving the transition position of train traction conditions. In [8], the solving capability of the algorithm was improved by improving genetic algorithm operators and introducing a reverse calculation operator, which saved the traction energy consumption of high-speed railway train operation while ensuring safety and punctuality. In [9], the authors proposed a fuzzy non-dominated sorting genetic algorithm to address the challenges posed by the manual driving characteristics and the variability in driving commands, aiming to optimize energy savings and establish an optimal energy consumption curve for trains. In [10], a novel driving strategy and energy consumption allocation-based urban rail train ATO control method was developed,

which significantly reduced traction energy consumption by extending the inertial coasting distance between stations. An Improved Quantum Genetic Algorithm (IQGA) is utilized in [11] to optimize energy consumption for urban rail transit's random traction strategies, resulting in a 4.06% energy reduction through a single-point random operation traction calculation model.

As hardware arithmetic power has improved, artificial intelligence algorithms, such as reinforcement learning algorithms, have achieved more admirable results in the fields of automatic control [12-14], intelligent driving [15] and online education [16, 17]. A traction control method for trains based on the LAG-LSTM model was proposed in [18], which predicts future traction conditions by extracting high-dimensional features from input data. This method was compared and analyzed with six other deep learning algorithms to verify its effectiveness. Additionally, [19] proposed intelligent train control algorithms based on expert systems and reinforcement learning algorithms, respectively. Comparative experiments revealed that the intelligent train control algorithm based on reinforcement learning algorithms performed better in terms of energy conservation. As outlined in [20] and [21], the train automatic driving methods under two reinforcement learning modes, policy-based and value function-based, were suggested, with the former simulating the train operation process. A subway train energy-saving driving method based on the DDPG algorithm was suggested in [22]. [23] studied the high-speed train speed curve optimization method under temporary speed limit conditions based on the DDQN algorithm and compared it with genetic algorithms. A collaborative deep reinforcement learning method oriented towards multi-objective parameter tuning was proposed in [24] to enhance the ability of agents to perceive and make adaptive decisions. A train energy-saving driving method based on the DQN algorithm was proposed in [25], with an analysis of the sensitivity of key hyperparameters in the algorithm. A subway train energy-saving operation control method based on the DDDQN algorithm was proposed in [26], which, while ensuring the safe and punctual operation of trains between stations, has better energy-saving benefits and convergence speed compared to differential evolution algorithms and the original DQN algorithm.

The extant research demonstrates the efficacy of reinforcement learning in addressing short-distance train operation control issues, such as those encountered in urban rail networks. However, the scope of reinforcement learning extends beyond short-distance applications, necessitating further research to enhance its performance in medium- and long-distance train operation control, including intercity and inter-provincial traffic management. This paper proposes a train operation control method based on DDQN, with the objective of enhancing the control performance of the ATO system. The primary objectives of this paper are as follows: Firstly, a high-speed train kinematics model will be constructed through analysis, and concomitantly, a simulation environment for train operation will be established. Secondly, a DDQN algorithm will be constructed to solve the optimal operating conditions of high-speed trains in the interval. This algorithm will then be compared with the traditional DQN algorithm to solve the optimal operating conditions of high-speed trains in the interval. Finally, the DDQN algorithm will be compared with the traditional DQN algorithm. The efficacy of the algorithm should be verified through a comparison and analysis with the traditional DQN algorithm. Thirdly, changes in the solution outcomes should be investigated under varying parameter configurations by manipulating the values of the hyperparameters.

II. Problem Description

The operation of a high-speed railway train involves a series of processes, including acceleration, cruising, and braking. These processes are normally executed in a specific sequence, known as the "full traction - uniform speed - full braking" mode. While this mode can result in the train reaching its destination within the scheduled time or even earlier, it also leads to significant energy consumption. Additionally, rapid acceleration can negatively impact the comfort of passengers. Therefore, it is critical to optimize the sequence of traction conditions to balance the trade-off between energy consumption and passenger comfort. The objective of optimizing train operation control is to achieve a balance between the time taken to run a train and the energy consumed. This is done by determining the optimal sequence of traction conditions, based on a given running time and distance, considering the gradient of the line and other conditions, with the aim of minimizing the energy consumption of the train throughout the entire operation. This is made in a way that ensures the train arrives at its destination on time and in a safe manner. The study is conducted by constructing a single-mass traction point for a high-speed train running on a restricted zone. The study presents a viable proposal for the energy-efficient operation of high-speed trains through the construction of a single-mass traction simulation model.

III. Model Construction

In accordance with Pontryagin's principle of extreme value, the optimal traction conditions for high-speed trains on the zone are comprised of four conditions: full traction, uniform cruise, idling, and full braking [2]. The body length of the high-speed train can be considered negligible in comparison with the spacing between high-speed railway stations. To reduce computational complexity, this paper proposes the construction of a single-mass model of the train and the conduction of a force analysis for the traction operation process. The relevant variables involved are shown in Table I.

The train kinematic equations are established as shown in (1):

$$\begin{cases} \dfrac{dt}{dv} = \dfrac{M}{F_r^j(v_i, x_i)} \\ \dfrac{dt}{dx} = \dfrac{1}{v} \end{cases} \tag{1}$$

To simplify the calculation, the study considers solely the traction energy consumption during train operation. The entire process of train operation is subdivided into a series

of sub-processes, with a time step of 1 s. Under this time step, the energy consumption of sub-processes can be regarded as the average value of the instantaneous power of the train.

TABLE I
PARAMETER DESCRIPTION

Symbol	Meaning	Unit
M	Train Traction Quality	t
g	gravitational acceleration	m/s^2
$F_r^j(v_i, x_i)$ $(j=0,1,2,3)$	The resultant force exerted on the train in different traction condition	kN
E_i^j $(j=0,1,2,3)$	Energy consumption of train operation in different traction condition	$kw \cdot h$
$F(v_i)$	Train output traction at speed v_i	kN
$f_{\omega_0}(v_i)$	Fundamental resistance per unit of train at moment i	N/kN
$f_{\omega'}(x_i)$	Additional resistance per unit of line to the train at moment i	N/kN
$B(v_i)$	Train output braking force at speed v_i	kN
A, B, C	Coefficients of the Davis equation	-
v_i	Instantaneous speed of the train at moment i	m/s
x_i	Position of the train at moment i	m
$i(x_i)$	Slope of line at x_i	
$R(x_i)$	Radius of line curve at x_i	m
$L_s(x_i)$	Length of tunnel at x_i	m
t_{use}	Actual train running time	s
t_{plan}	Planned Train Running Time	s
v_0	Train speed at the initial moment	m/s
$v_{t_{plan}}$	Train terminal speed	m/s
$v_i^{\lim}$	Speed limit for the current time zone	m/s
$F_{\max}$	Maximum traction output of the train	kN
$B_{\max}$	Maximum braking power output of the train	kN
Δx	The gap between x_a and x_s	m
x_a	Actual operating distance	m
x_s	Total distance between stations	m
μ_i (i=1,2,3)	weighting factor	-

The energy consumption of the entire process of train operation is the sum of sub-processes, and the energy consumption of traction operation is primarily divided into the following four categories according to the different working conditions of the train:

1. The acceleration phase at the commencement of train operation, when energy consumption can be regarded as the increase in the speed of train operation and the work done to overcome resistance, and the formula for calculating the combined force and energy consumption corresponding to this phase are shown in (2) and (3):

2. The train continues to travel at a constant velocity during the phase in which the energy expenditure is solely that required to surmount the resistance. The formulae for calculating the aggregate force and energy expenditure corresponding to this phase are presented in (4) and (5), respectively.:

$$\begin{aligned} F_r^0(v_i, x_i) &= F(v_i) - \frac{Mg}{1000}(f_{\omega_0}(v_i) + f_{\omega'}(x_i)) \\ &= F(v_i) - \frac{Mg}{1000}(A + Bv_i + Cv_i^2 \\ &\quad + i(x_i) + \frac{600}{R(x_i)} + 0.00013L_s(x_i)) \end{aligned} \tag{2}$$

$$E_i^0 = \sum_{i=0}^{u} \frac{1}{2}(F(v_i) \cdot v_i + F(v_i) \cdot (v_i + a_i)) \tag{3}$$

$$F_r^1(v_i, x_i) = 0 \tag{4}$$

$$E_i^1 = \sum_{i=0}^{u} F(v_i) \cdot v_i \tag{5}$$

3. The following formulas represent the calculation of the combined forces and energies consumed during the train's idling phase, a period in which the energy consumption of the train's operation is zero. the corresponding combined force and energy consumption calculation formulas for this phase are shown in (6) and (7):

$$\begin{aligned} F_r^2(v_i, x_i) &= -\frac{Mg}{1000}(f_{\omega_0}(v_i) + f_{\omega'}(x_i)) \\ &= -\frac{Mg}{1000}(A + Bv_i + Cv_i^2 \\ &\quad + i(x_i) + \frac{600}{R(x_i)} + 0.00013L_s(x_i)) \end{aligned} \tag{6}$$

$$E_i^2 = 0 \tag{7}$$

4. Train deceleration braking stage, this study only explores the train operation energy consumption in the case of a single train, without considering the regenerative braking energy calculation, so the train operation energy consumption in this stage is also regarded as 0. This stage corresponds to the combined force and energy consumption calculations as shown in (8) and (9):

$$\begin{aligned} F_r^3(v_i, x_i) &= -B(v_i) - \frac{Mg}{1000}(f_{\omega_0}(v_i) + f_{\omega'}(x_i)) \\ &= -B(v_i) - \frac{Mg}{1000}(A + Bv_i + Cv_i^2 \\ &\quad + i(x_i) + \frac{600}{R(x_i)} + 0.00013L_s(x_i)) \end{aligned} \tag{8}$$

$$E_i^3 = 0 \tag{9}$$

The traction and braking forces corresponding to the output of the train at different speeds are calculated as shown in (10) and (11):

$$F(v_i) = \begin{cases} -1.026 \bullet v_i + 300 & v_i \in [0, 33.06] \\ 8750/v_i & v_i \in [33.06, 83.33] \end{cases} \tag{10}$$

$$B(v_i) = \begin{cases} 215.28 \bullet v_i & v_i \in [0, 1.39] \\ -1.026 \bullet v_i + 300 & v_i \in [1.39, 29.64] \\ 8022.22/v_i & v_i \in [29.64, 83.33] \end{cases} \tag{11}$$

In this study, the objective function is constructed as shown in (12) by considering the energy consumption of train operation, the gap between the actual train operation time and the scheduled time, and the accuracy of stopping.

It is imperative to incorporate a training program encompassing traction operation, adhering to the constraints delineated in (13). The initial and terminal moments of operation are to be conducted at a velocity of 0, ensuring the

safety of the operation. The operational speed of the train should not exceed the designated speed limit of the designated zone. The process of output operation necessitates adherence to the stipulated constraints. It is imperative to emphasize that the traction and braking forces must not exceed the maximum output capacity. To ensure the punctuality and accuracy of the train, the error in the stopping position should be controlled within 0.5 m, and the running time should be controlled within 3%.

$$\min \quad \mu_1 \bullet \sum_{j=0}^{3} E^j + \mu_2 \bullet \left| t_{use} - t_{plan} \right| + \mu_3 \bullet \Delta x \tag{12}$$

$$\begin{cases} v_0 = v_{t_{plan}} = 0 \\ v_i < v_i^{\lim} \quad i \in (0, t_{plan}) \\ F \le F_{\max}, B \le B_{\max} \\ \Delta x = \left| x_a - x_s \right| < 0.5 \\ \dfrac{\left| t_{use} - t_{plan} \right|}{t_{plan}} \bullet 100\% < 3\% \end{cases} \tag{13}$$

IV. Algorithm Design

A. Design of reinforcement learning framework

In the context of high-speed train operation, the on-board controller is responsible for reading the current moment and the current position, in addition to the line state information, train speed, and train running time. It then calculates the traction conditions that should be taken by the train at the next moment. The entire process of high-speed train operation can be conceptualized as a finite Markov decision-making process, for which reinforcement of learning-related algorithms can be employed to solve the problem. The on-board controller of the high-speed train can be regarded as an intelligent body, and in this paper, the current position of the train, the current speed of the train, and the running time of the train are three states s. To simplify the computation process of the problem, this paper discredited the whole process of the train operation, The time step is set to a constant value, designated as Δt. To ensure the integrity of the model and its generalizability, the current position and speed of the train are set as the time step, thereby ensuring that the order of magnitude difference between these variables does not adversely impact the model's training or its generalizability. Consequently, the current position of the train is assigned to the running time, ensuring that the order of magnitude processing is accurately reflected. The state $\boldsymbol{s}$ in this study can be expressed as (14):

$$\boldsymbol{s} = \left(\theta_1 \bullet d,\ v,\ \theta_2 \bullet t \right) \tag{14}$$

In this study, the variables d, v, t, and order-of-magnitude processing parameters were used to analyze the operation of a high-speed train. The order-of-magnitude of the train's running distance was set at 10^4, and the running time order-of-magnitude was calculated. of-magnitude of 10^2 to 10^3, to eliminate the state order-of-magnitude of the disparity between the neural network training caused by the adverse effects of the large, in this paper the two parameters were taken as 0.001 and 0.01.

According to the current input state, the on-board controller selects and outputs the train traction condition command, calculates the combined force on the train at the current moment by combining the selected condition command with the current speed of the train, and then calculates the speed change value and the running distance of the train within the time step, to obtain the state of the next moment. The output traction condition command is the action selected by the intelligent body, and this paper adopts the construction of discrete action space A as (15):

$$A = \left\{ a_i \mid \ a_i = 0,\ 1,\ 2,\ 3 \right\} \tag{15}$$

In this context, 0 denotes maximum traction, 1 corresponds to constant speed cruise, 2 corresponds to idling, and 3 corresponds to maximum braking. In order to address the actual situation and reduce the complexity of problem resolution, it is established that the resonant force traction condition and full force braking condition cannot be converted to each other directly. The objective of reinforcement learning is to derive an optimal strategy to maximize the sum of rewards. The optimization objective of high-speed train traction control should comprehensively reflect the safety, punctuality, stopping position accuracy and energy saving of the train during operation. The process of constructing the reward function is as follows: 1) In the event that the on-board controller manipulates the train during operation in such a manner that the current speed of the train exceeds the speed limit of the line corresponding to the current position, or if the train's actual running time exceeds the maximum time of the train operation as limited by the algorithm, then The reward value is set to -P, and the reward value is set to -P, and the reward value is set to -P. The value is set to -P, and at the same time this training round is terminated, and the program is initialized to start the next training round. If the actual running distance of the train is greater than the total distance between stations, the following calculations are to be made: the total running time of the train, the speed when it exceeds the end position of the line, the difference between the actual running distance and the total distance between stations, and the energy consumption value. These calculations are to be weighted and summed, and the training round is to be terminated at the same time. In other states, the instant reward is set to a fixed value of -1. In summary, the reward function r is constructed as (16):

$$r = \begin{cases} -P \quad , v_t > V_{t-\lim} \\ -P \quad , t_{use} \ge T_{\max} \\ P - (w_1 \bullet \log t_{use} + w_2 \bullet v_{end} + w_3 \bullet \Delta d_{end} + w_4 \bullet E_{use}) \quad , x \ge x_s \\ -1 \quad , otherwise \end{cases} \tag{16}$$

In the formula, P is a fixed value used to control the final value of the reward function size. This paper takes 100 for the moment of t train speed; for the moment of t train running in the location of the speed limit; for the final moment of the train speed; for the train's actual running time; for the algorithm to limit the maximum time of the train; for the final moment of the train running between the location of the total distance between the station; for the final moment of the train running, the total energy consumption; respectively for the running time, end speed, end position difference and running energy consumption corresponding to the weight coefficient, this paper based on the expert scoring method, taking into account the size of each numerical order of magnitude, and ultimately determined that the four weight coefficients were valued at 3, 3, 0.1, 0.01, respectively.

B. Algorithm design for this model

The Double Deep Q-Network (DDQN) algorithm represents a significant advancement in the field of deep reinforcement learning, building upon the foundations of the Deep Q-Network (DQN) algorithm. Like the DQN algorithm, the DDQN algorithm employs a neural network to approximate the value function. The fundamental structure of the DDQN algorithm is depicted in Figure 1. During the training phase, the DDQN algorithm incorporates the experience replay mechanism (Experience Replay), ensuring the accuracy of the training outcomes. The continuous interaction between the intelligent agent and its environment generates a sequence of state transition trajectories, τ, as outlined in equation 17:

$$\tau = (s,\ a,\ r,\ s',\ b) \tag{17}$$

Where s denotes the current moment state; a denotes the action selected at the current moment; r denotes the immediate reward obtained at the current moment; s' denotes the next moment state; and b denotes a Boolean variable that determines whether state s' is a termination state.

The trajectory data obtained is stored in the experience pool, and during the weight update process of the neural network, a portion of the trajectory data is sampled from the experience pool as training samples. The purpose of experiencing playback is threefold: firstly, to break the correlation between the data; secondly, to find accurate gradient estimates; and thirdly, to improve the utilization of experience. In the context of reinforcement learning algorithm training, the state transfer process corresponding to a single occurrence of the reward value is more reflective of the changes in the environment in which the intelligent body is located compared to the process that occurs multiple times. Consequently, it is necessary to extract this process separately for the training of the intelligent body. To enable the model to fully learn from the experience of different reward tiers, thus improving the stability and generalization ability of the model, this paper adopts the stratified sampling method. Following the intelligent body's interaction with the environment and the collection and storage of a specified number of experiences in the experience playback pool, the reward function value of each experience is divided by a constant value, N, to generate a new list. This list serves as the label for stratified sampling, and the samples are then extracted from the experience playback pool based on the distribution of values in this label. The specific process is outlined below: Initially, the experiences are arranged according to their reward value, and the reward value is divided by a fixed value to generate a new list. The Counter function is then utilized to enumerate the number of occurrences of each element in the new list. The determination is based on the distribution of values in this list. The elements that appear only a single time and those that appear many times and count the elements that appear only a single time. If the number reaches the batch size, then sample the elements that appear only a single time from the experience playback pool. If the number reaches batch size, then the experience corresponding to the element with single occurrence is taken from the experience playback pool to form the training sample, otherwise the experience corresponding to the element with multiple occurrences is randomly sampled to form the training sample.

In the context of traditional deep reinforcement learning methodologies, a single value function network is typically implemented for the purpose of fitting the value function and selecting actions. However, this approach often results in an overestimation of the state-action value function by the intelligence. To address this limitation, a novel target network is incorporated into the original algorithm, with the specific objective of evaluating the Q-value. The traditional deep reinforcement learning algorithm selects actions and calculates Q-values in the following way, as shown in (18) and (19):

$$a^* = \arg\max_{a \in A} Q(s_{t+1}, a, \omega) \tag{18}$$

$$Q(s_t, a_t, \omega) = r_t + \max_a Q(s_{t+1}, a^*, \omega) \tag{19}$$

And in this algorithm, the Q value is calculated as shown in (20):

$$Q(s_t, a_t, \omega^-) = r_t + \max_a Q(s_{t+1}, a^*, \omega^-) \tag{20}$$

After a number of rounds of weight updating in the main network, the target network is synchronized with each weight parameter of the main network, and the weight parameters of the target network are kept unchanged for the rest of the time, so as to reduce the error of the intelligent body's evaluation of the action value function, in order to obtain a predicted value that is closer to the true value. When the intelligent body selects an action, if it selects the action corresponding to the highest Q value in the current state based on the greedy strategy, it may miss the action that is better from the global perspective. In order to improve the possibility of the algorithm to find the global optimal solution, this paper adopts the ε -greedy strategy to improve the search breadth, which can be expressed in (21):

$$a_t = \begin{cases} \arg\max_a Q(s_t, a_t, \omega) & p = 1-\varepsilon \\ \underset{a \in A}{random} & p = \varepsilon \end{cases} \tag{21}$$

where ε denotes the selection probability. To prevent ε from being too small to cause the algorithm to be too greedy and reduce the exploration breadth, or too large to cause the algorithm to be too random, this paper sets the initial value of ε, the minimum value of ε_{min}, and $\Delta\varepsilon$, which means the decay value of ε, so that ε value decreases linearly in the training process.

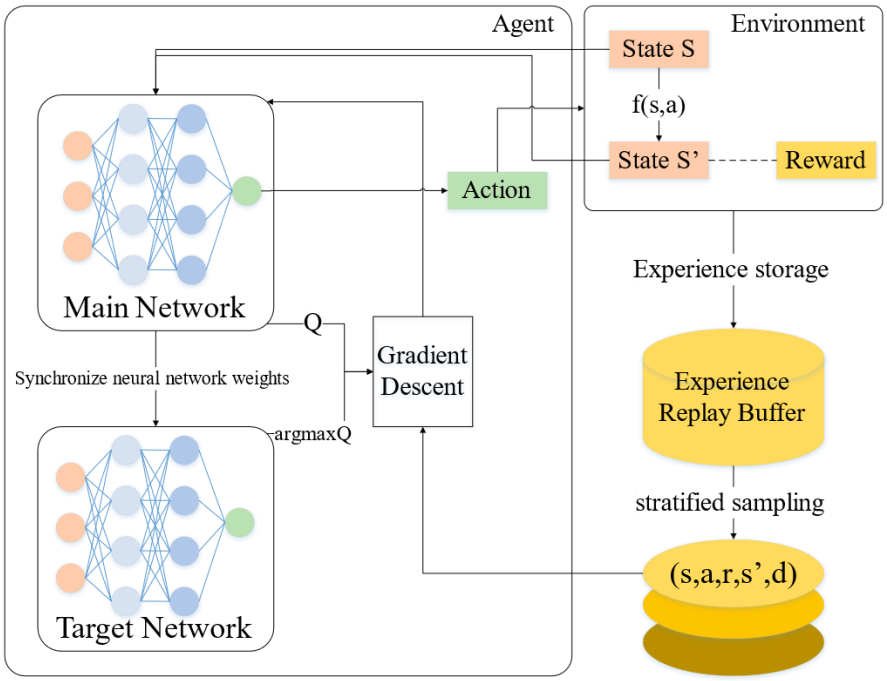

Fig.1 Basic structure of DDQN algorithm

In summary, the DDQN algorithm with its pseudo-code can be represented in the following form:

Train traction control optimization algorithm based on DDQN

Step1: Algorithmic input:

- Define the neural network model, including its structure, optimizers used, etc. ;
- Defines algorithm-related parameters: learning rate α, number of training num_episodes, ε-greedy policy selection probability and associated values ε, εmin, $\Delta\varepsilon$, experience replay buffer size buffer_size, experience replay pool size during training of neural network buffer_size, sampling batch size batch _size, discount factor γ;
- Define the relevant data required for the training environment, such as line data, vehicle data, planning running time, running distance, etc.; Define the state transfer rules and the calculation method related to the combined force applied to the train;

Step2: Initialization:

- Initialize the primary and target networks using random parameters.
- Initializing the experience playback pool.

Step3: Model Training:

```
for episode in range(num_episodes):
    Initialize the state of the train intelligences as s0=(0,0,0)
    done = false
    while not done:
        selection of actions by ε-greedy strategy a_t ;
        The working conditions correspond to the execution of the action, according to the combined force calculation method to calculate the combined force and acceleration value of the train, so as to deduce the initial speed and position of the train at the next moment ;
        Actual running time of the train t += 1;
        Generate the next moment state. Calculate the immediate reward value r (see (6) for calculation)
        Determine done == True or False according to the state transfer rules
            if done = True
                End this episode and restart the loop
            else
                Continue this episode
        Store the state transfer trace τ
        Update the current state to s_{t+1}
        if the number of state transfer trajectories τ stored in the experience playback pool > buffer_size:
            Perform hierarchical sampling of state transfer trajectories in the experience playback pool;
            Calculate Q-value and target Q-value from the main network and target network, respectively;
            Calculate loss function values;
            Backpropagation to update the main network model;
            Synchronize target and master network parameters every N steps;
        Update the ε value
```

Step4: Model Testing:

- Save the neural network model after training.
- Given the test environment. ,

```
Initialize the train intelligence state as s0=(0,0,0)
done = false
while not done:
    Load the trained neural network model
    Input the current state and output the action
    Record and execute the traction condition corresponding to the action and update the state
    Determine if done = True or False according to the state transfer rule.
    if done = True
        End the loop and output the sequence of conditions, total running time and total energy consumption.
    else
        Continue the loop
```

V. Case Study

A. Environment Setup

The interstation slope data is shown in Fig. 2

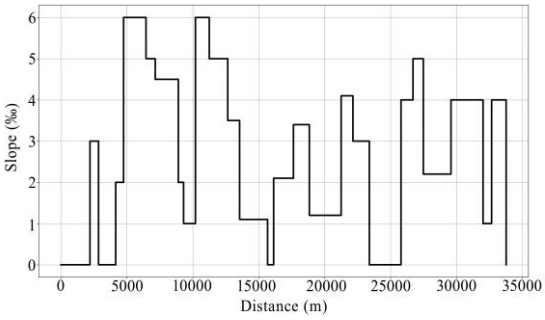

Fig.2 Slope data between stations

TABLE II
TRAIN-RELATED PARAMETERS AND MAIN LINE DATA

Symbol	Meaning	parameter size	Unit
M	Train Traction Quality	428	t
A	Coefficients of the Davis equation	0.66	-
B	Coefficients of the Davis equation	0.00245	-
C	Coefficients of the Davis equation	0.000132	-
t_{plan}	Planned Train Running Time	590	s
$v_i^{\lim}$	Speed limit for the current time zone	83.33	m/s
$F_{\max}$	Maximum traction output of the train	300	kN
$B_{\max}$	Maximum braking power output of the train	299	kN
x_s	Total distance between stations	33770	m

The experimental equipment setup is displayed in Table III and the relevant parameters involved in the algorithm are shown in Table IV.

TABLE III
EXPERIMENTAL EQUIPMENT SETUP INFORMATION

Hardware/Frameworks	Parameters and Versions
CPU	Intel Core i5-12500H
GPU	NVIDIA RTX 3060Laptop
RAM	16 GB
Operating System	Windows 11
Program Language	Python 3.9
ML Library	Pytorch

TABLE IV
DDQN ALGORITHM HYPERPARAMETER VALUES

Hyperparameter	Meaning	Parameter size
num_episodes	Number of model training sessions	2500
T	Algorithms to limit maximum train running time	70000
α	Learning rate	0.001
ε	Initial selection probability	1
ε_{min}	Final choice probability	0.01
$\Delta\varepsilon$	Selection of probability decay value per step	1/1900
buffer_size	Experience playback buffer size	100000
batch_size	Batch size per sample	32
γ	discount factor	0.99
N	Synchronizing the number of steps between the target network and the main network parameters	100
	Optimizer	Adam

To verify the effectiveness of the algorithm, and to explore the impact of hyperparameters on the performance of the algorithm, in this section, a simulation case is designed based on the actual line and operation data of the interval from station A to station B in the Qinhuangdao-Shenyang High-speed Railway. The train type in the case is CRH3-350, and the relevant parameters of the train and the main line data are shown in Table II, Due to the lack of comprehensive data in the relevant public information, we have only obtained the slope data of this section of the line for calculating the additional drag force on trains.

B. Training and Testing Setup

1) Comparative analysis of algorithm training results

To evaluate the training effect of the algorithms, we collected the traction operation results of manual driving by train drivers in real scenarios and compared the traction operation results generated by the two algorithms, DQN and DDQN, with those of manual driving in terms of energy consumption, the distance gap, and the runtime gap.

2) Analysis of the impact of different hyperparameters on the algorithm training results

To investigate how the hyperparameters in the algorithm affect the training results of the algorithm, we selected two hyperparameters, the learning rate α and the ε decay value $\Delta\varepsilon$ and compared and analyzed the generated traction operation results in terms of energy consumption, the distance gap, and the runtime gap.

The evaluation indicators involved in these two experiments are calculated as follows.

■ Energy consumption for traction operation

In the comparative analysis of the training results, we use the manual driving results as a reference term, which can be expressed in (22):

$$\varphi_E = \frac{E_{al}}{E_m} \times 100\% \tag{22}$$

In the hyperparametric impact analysis we selected the result with the largest combined reward value as the reference term, as shown in (23):

$$\varphi_E = \frac{E_{al}}{E_{al\_\max R}} \times 100\% \tag{23}$$

The smaller the value of this indicator, the better the results of the model in terms of energy consumption.

■ Distance gap

In the above two experiments, the ratio of the gap between x_a and x_s to x_s was used as an indicator for the assessment of this gap, which can be expressed in (24):

$$\varphi_D = \lg(|x_a - x_s| + 1) \times 100\% \tag{24}$$

The closer the value of this indicator is to 0%, the better the model solution is in terms of stop location accuracy.

■ Runtime gap

In the above two experiments, the ratio of the gap between t_{use} and t_{plan} to t_{plan} is used as the index for the evaluation of this error, and the specific formula is shown in (25).

$$\varphi_T = \frac{t_{use}}{t_{plan}} \times 100\% \tag{25}$$

The closer the value of this indicator is to 100 %, the better the model's solution in terms of punctuality.

C. Model solution and analysis

Based on the proposed DDQN train traction control optimization algorithm, the optimal speed profile of a train under a given scenario is calculated and compared with the optimal speed profile obtained by the traditional DQN algorithm, and the on-time performance, energy consumption and other indexes are compared, to verify the

feasibility and validity of the proposed method. Fig. 3 reflects the change of the average reward value with the number of iterations between the traditional DQN algorithm and the DDQN algorithm, and the train speed-distance curves obtained by the two algorithms are shown in Fig. 4, and the specific experimental results are shown in Table V.

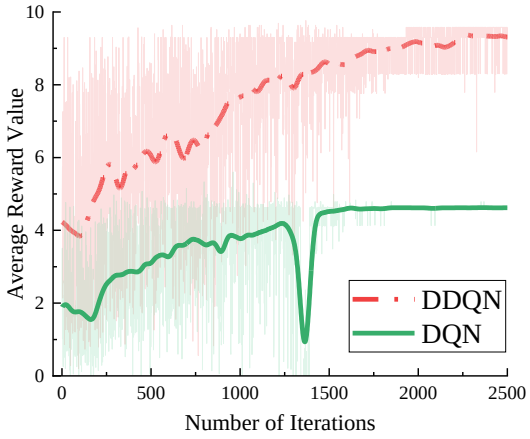

Fig.3 Plot of variation in average reward value of DQN and DDQN algorithm

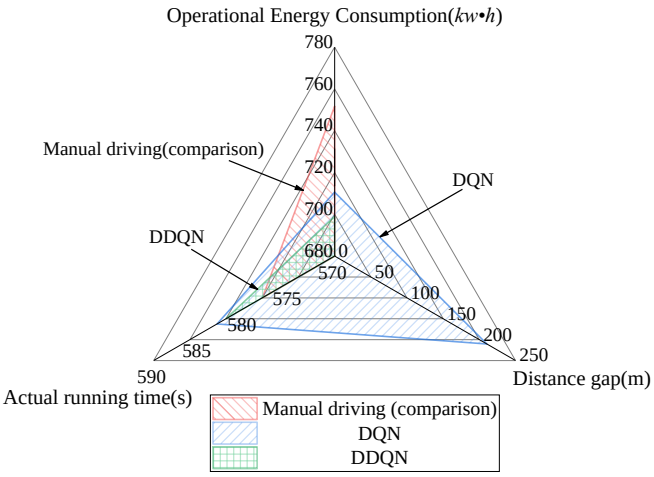

Fig.5 The actual performance of the manual driving mode, the DQN algorithm and the DDQN algorithm

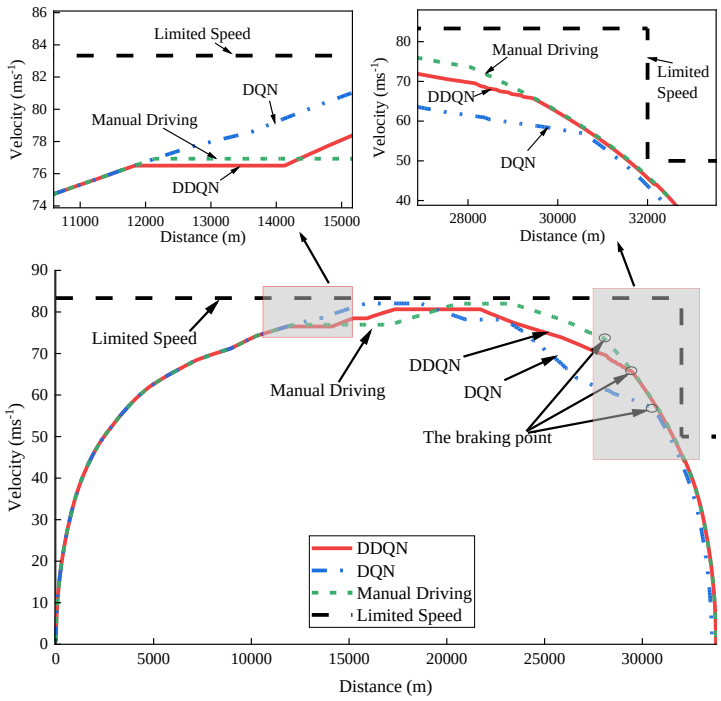

Fig.4 Train speed-distance curves obtained by DQN algorithm and DDQN algorithm

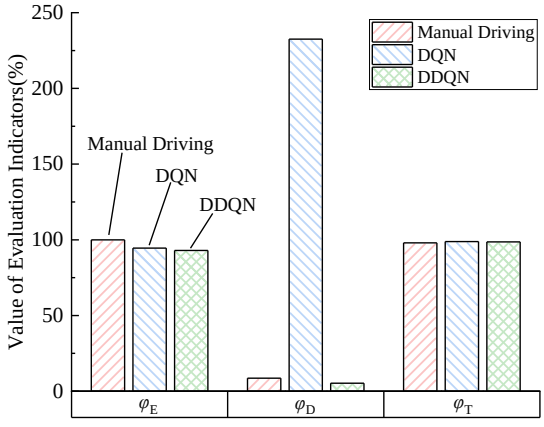

Fig.6 Statistics of evaluation indicators for different algorithms

TABLE V
COMPARISON OF EXPERIMENTAL RESULTS BETWEEN DQN ALGORITHM AND DDQN ALGORITHM

Algorithm name	Energy Consumption (kW•h)	Distance(gap) (m)	Runtime(s)	Convergence value of the reward function
Manual driving	751.64	33769.78(0.22)	578	—
DQN	710.60	33559.50 (210.50)	583	4.62
DDQN	698.86	33770.13(0.13)	582	9.23

The actual performance of the manual driving mode, the DQN algorithm and the DDQN algorithm are shown in the Fig.5 and Table V, and the performance of the different traction driving strategy methods in terms of evaluation

indicators is shown in Fig.6. As can be seen from the data in Table V, the manual driving mode, the DQN algorithm used for comparison, and the DDQN algorithm used in this paper are all within the specified running time in terms of running time, and the DDQN algorithm is optimal in terms of running energy consumption and running distance error. In terms of running energy consumption, the DDQN algorithm saves 11.74 kW•h compared to the DQN algorithm and saves 52.78 kW•h compared to manual driving. Compared to the manual driving mode and the traditional DQN algorithm, the train intelligence controlled by the DDQN algorithm extends the idle time by increasing the number of idle runs, which reduces the traction work time under the premise of ensuring on-time train operation, thus saving traction energy consumption.

Comparison of traditional DQN and DDQN algorithms for the training of the average reward value changes can be seen, both algorithms in the training of the initial period of large fluctuations, but with the increase in the number of times of training, the average reward value gradually tends to stabilize, and the DDQN compared to the DQN algorithm in the convergence of the speed of convergence is almost the same, the convergence of the average reward value of the DQN algorithm is about two times. It is proved that this algorithm can find a better solution compared to the DQN algorithm.

D. Sensitivity analysis

This section explores the impact of the algorithm's key hyperparameters on the model training situation. The focus is on two key hyperparameters, the learning rate α and the ε decay value $\Delta\varepsilon$. The former determines the step size of the update from the current function value to the target function value at each update step, and the latter determines whether the model is biased towards exploring a better solution or utilizing current experience. By keeping the neural network structure and other hyperparameters unchanged, the effect of different hyperparameters sizes on the change in reward function values can be observed.

TABLE VI
RESULTS CORRESPONDING TO DIFFERENT SIZES OF LEARNING RATE α

α	Energy Consumption (kW•h)	Distance(gap) (m)	Runtime(s)	Convergence value of the reward function
0.1	952.50	34086.99(316.99)	504	9.08
0.01	904.96	33949.72(179.72)	504	9.30
0.001	736.03	33760.79(9.21)	568	9.79
0.0001	726.86	34059.10(289.1)	562	7.08

Fig.8 shows the variation of the average value of the reward function value with increasing number of iterations, given different sizes of learning rates. Fig.7 shows the final velocity-distance curve derived from the corresponding model. The actual performance of different learning rates is shown in Fig.9, and evaluation indicators of the algorithm at different learning rates are shown in Fig.10. Combining Fig.8, Fig.9 and Table VI for the problems required to be solved in this paper, the neural network model works best when the learning rate is of the order of 0.001. If the value of the learning rate is too large, due to the large update step, the model training results cannot be stabilized near the optimal solution, and the average reward function value fluctuates more; if the learning rate is too small, although the average value of the reward tends to converge, the model will fall into the local optimal solution.

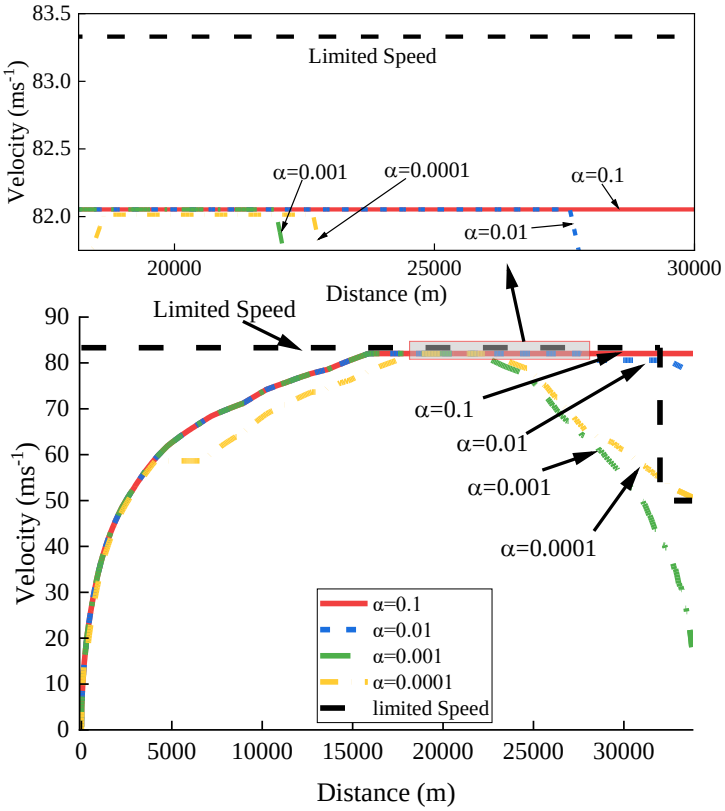

Fig. 7 Velocity-distance curves for different learning rates.

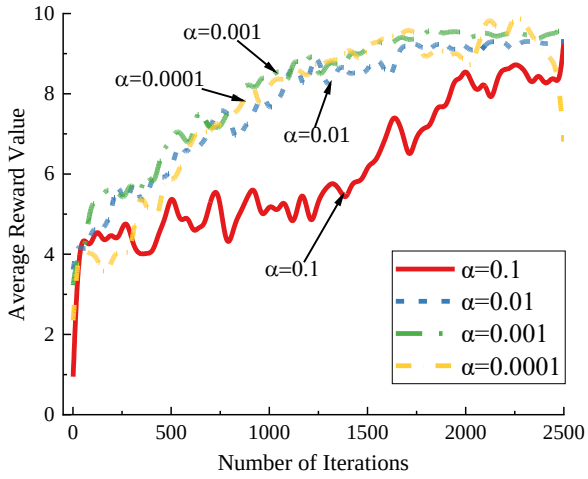

Fig. 8 Variation of average reward function value for different size learning rate α

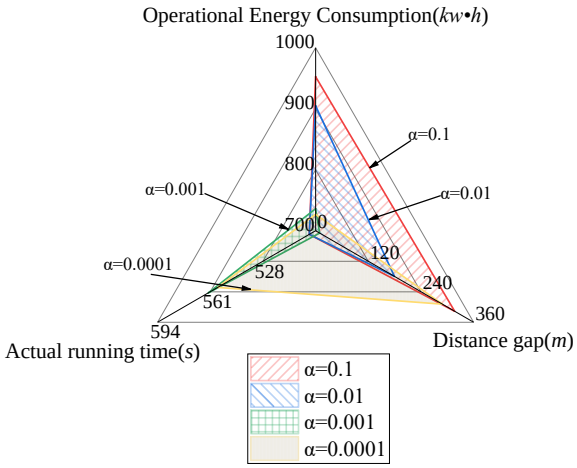

Fig.9 The actual performance of different sizes of learning rate α

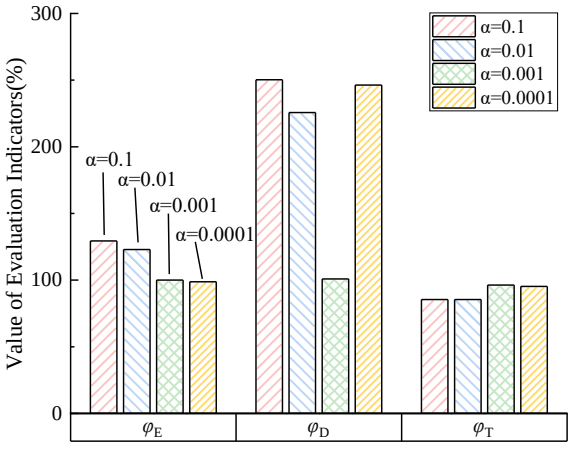

Fig.10 Statistics on evaluation indicators at different learning rates

Fig.11 shows the variation of the average value of the reward function value with increasing number of iterations for given values of $\Delta\varepsilon$ of different magnitudes. Fig.12 shows the velocity-distance curves derived from the corresponding model. The actual performance for different $\Delta\varepsilon$ is shown in Fig.13, and the evaluation indicators of the algorithm for different values of $\Delta\varepsilon$ are shown in Fig.14. Combined with Fig.11 , Fig.13 and Table VII, it can be seen that for a given number of iterations of the neural network model, the $\Delta\varepsilon$ value of about 1/2500 is relatively the best effect; the larger the $\Delta\varepsilon$ value, the shorter the model's exploration of the environment, it will soon enter the learning mode using the existing optimal experience, and the faster the convergence, but the training effect is poor; and the smaller the $\Delta\varepsilon$ value, the change in the reward function value can be seen from its average reward value is increasing, but in the case of a given number of iterations has not converged, if a larger number of iterations of the model is given, then the model training time will increase accordingly, affecting the training efficiency.

TABLE VII
RESULTS CORRESPONDING TO DIFFERENT $\Delta\varepsilon$ VALUES

$\Delta\varepsilon$	Energy Consumption (kW•h)	Distance(gap) (m)	Runtime(s)	Convergence value of the reward function
1/625	696.63	34107.61(337.61)	575	8.07
1/1250	727.27	33438.12(331.88)	549	8.49
1/2500	728.82	33736.41(33.59)	580	9.30
1/5000	749.90	33664.10(105.9)	566	7.21

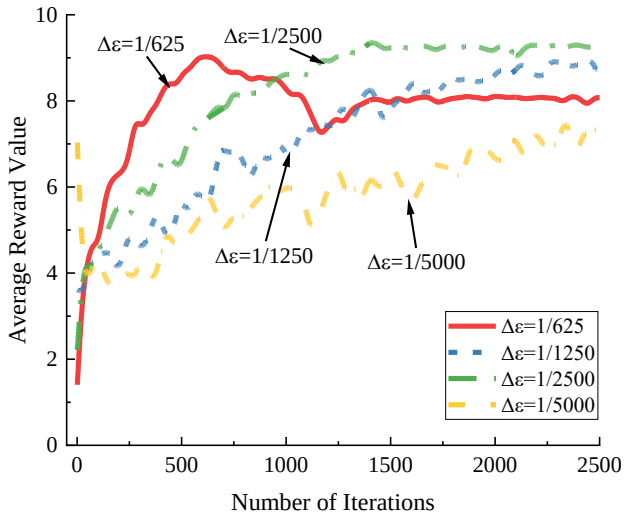

Fig. 11 Variation of average reward function values for different sizes of $\Delta\varepsilon$ values

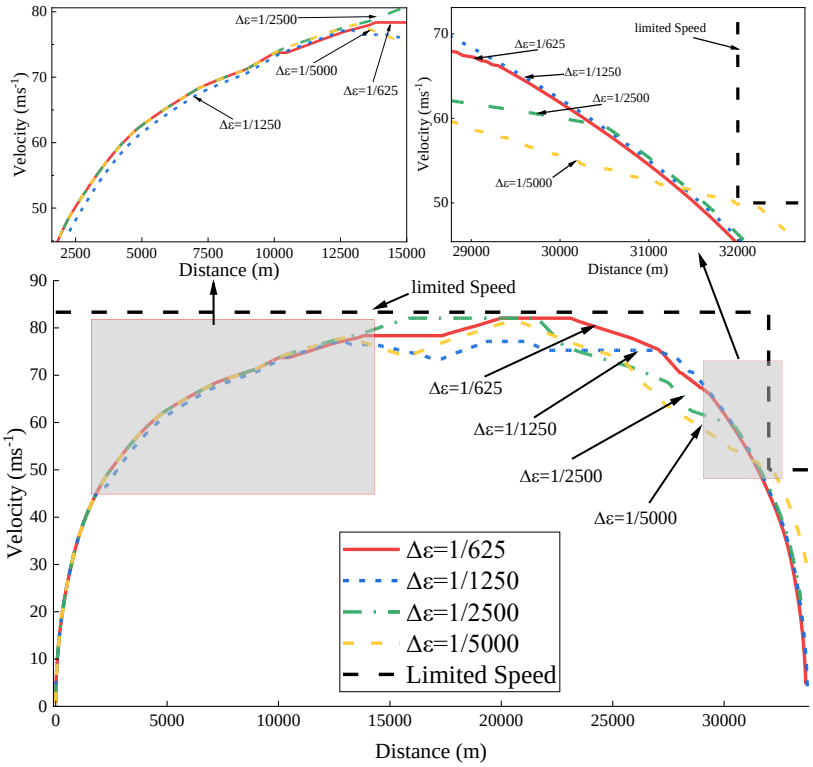

Fig. 12 Velocity-distance curves for different values of $\Delta\varepsilon$

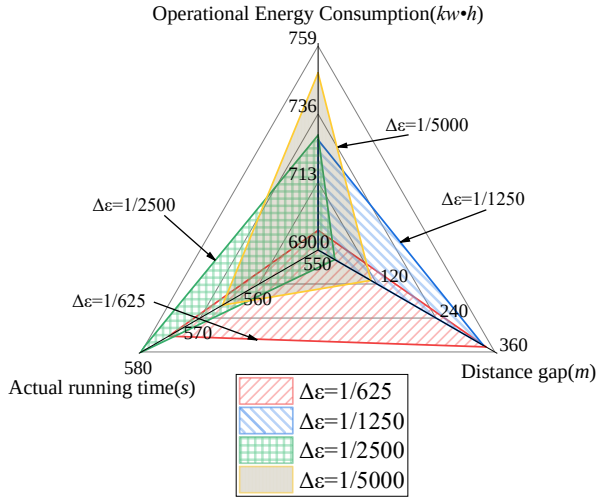

Fig.13 The actual performance of different $\Delta\varepsilon$

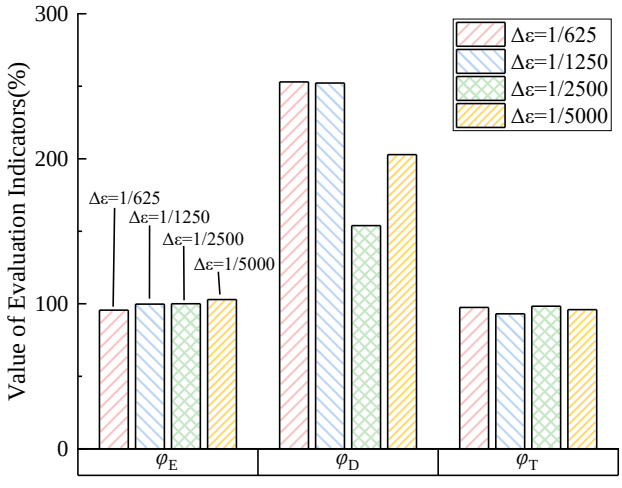

Fig.14 Statistics on evaluation indicators at different $\Delta\varepsilon$

V. Conclusions

In this study, an optimization method based on the DDQN algorithm is proposed to address the traction control problem of high - speed railway trains, considering energy consumption, safety and punctuality.

Simulation experiments using actual line data from a specific section show that the driving strategy obtained by this algorithm ensures that trains arrive at their destinations within the specified running time. It also makes full use of idle conditions to improve the utilization rate of traction energy consumption. Compared to the traditional DQN algorithm, it can save 11.74 kW•h of energy traction.

By controlling the learning rate α and $\Delta\varepsilon$, it is found that the best model training effect is achieved when the learning rate α is on the order of 0.001, and the sequence of traction conditions obtained is closest to the optimal sequence. When the $\Delta\varepsilon$ value is set at 1/2500, the model is most effective for a given number of training times. There is still room for improvement in the existing research, such as balancing the relationship between model training time and convergence results. The proposed algorithm has good application prospects in solving traditional problems in the railway industry.

References

[1] T. Chen, “Traction Energy Consumption Measuring Methods Study and Quantification Analysis on Energy Impact Factors of High-Speed Train,” M.S. thesis, Beijing Jiaotong University, Beijing, 2011.

[2] P. Howlett, "An optimal strategy for the control of a train," The Journal of the Australian Mathematical Society. Series B. Applied Mathematics, vol. 31, pp. 454-471, 1990.

[3] P. Wang, A. Trivella, R. M. P. Goverde, F. Corman, "Train trajectory optimization for improved on-time arrival under parametric uncertainty," Transportation Research Part C: Emerging Technologies, vol. 119, p. 102680, 2020.

[4] A. Cunillera, A. Fernández-Rodríguez, A. P. Cucala, A. Fernández-Cardador, M. C. Falvo, "Assessment of the Worthwhileness of Efficient Driving in Railway Systems with High-Receptivity Power Supplies," Energies, vol. 13, pp. 1836, 2020.
[5] W. Li, S. Zhao, K. Li, Y Xing, Q Li, W Yao, "GSOANR-based multi-objective train trajectory optimization," International Journal of Rail Transportation, vol. 12, pp. 733-748, 2023.
[6] M. Tang and Q. Wang, "Research on energy-saving optimization of EMU trains based on golden ratio genetic algorithm," Journal of Railway Science and Engineering, vol. 17, no. 01, pp. 16-24, 2020.
[7] X. Yan, B. Cai, B. Ning, W. Shangguan, "Research on multi-objective high-speed train operation optimization based on differential evolution," Journal of the China Railway Society, vol. 35, no. 09, pp. 65-71, 2013.
[8] J. Li and G. Chen, "Energy saving optimization of high-speed train based on improved genetic algorithm," Railway Computer Application, vol. 30, no. 03, pp. 5-9, 2021.
[9] Y. Pan and Z. Fu, "Energy-saving optimization of EMU trains considering the manual driving," Journal of Railway Science and Engineering, vol. 18, no. 05, pp. 1105-1112, 2021.
[10] S. Su and T. Tang, "Optimal train control for ATO system," Journal of the China Railway Society, vol. 36, no. 12, pp. 50-55, 2014.
[11] D. Li, X. Meng, Z. Han, S. Xu, B. Zhang, L. An, R. Wang, "Research on Energy Saving Optimization of Random Traction Strategy for Urban Rail Transit," Engineering Letters, vol. 31, no.1, pp.287-294, 2023
[12] F. Wang, Y. Chen, S. Yuan, et al. "Robot path planning based on self-attention mechanism combined with DDPG," Computer Engineering and Applications, 2023, 1-12. Available: https://link.cnki.net/urlid/11.2127.tp.20230920.0937.010 (Accessed on: Oct. 13, 2024). DOI: 10.3778/j.issn.1002-8331.2307-0009.
[13] J. Ren, Y. Liu, X. Hu, C. Xiang, X. Luo. "Motion planning model for autonomous driving in complex traffic scenarios," Computer Engineering and Applications, vol. 60, no. 15, pp. 91-100, 2024.
[14] Y. Gao, L. Ren, T. Shi, T. Xu, J. Ding, "Autonomous Obstacle Avoidance Algorithm for Unmanned Aerial Vehicles Based on Deep Reinforcement Learning," Engineering Letters, vol. 32, no. 3, pp.650-660, 2024
[15] T. Sun, C. Ma, Z. Li, K. Yang, "Cloud Computing-based Parallel Deep Reinforcement Learning Energy Management Strategy for Connected PHEVs," Engineering Letters, vol. 32, no. 6, pp.1210-1220, 2024.
[16] S. Yu, J. Li, T. Zhang, "Exercise Recommendation Algorithm Based on Reinforcement Learning," Engineering Letters, vol. 32, no. 10, pp.1947-1956, 2024
[17] J. Li, S. Yu, T. Zhang, "Learning Path Recommendation Based on Reinforcement Learning," Engineering Letters, vol. 32, no. 9, pp.1823-1832, 2024
[18] J. Yin, C. Ning, T. Tang, "Data-driven models for train control dynamics in high-speed railways: LAG-LSTM for train trajectory prediction," Information Sciences, vol. 600, pp. 377-400, 2022.
[19] J. Yin, D. Chen, L. Li, "Intelligent Train Operation Algorithms for Subway by Expert System and Reinforcement Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 15, pp. 2561-2571, 2014.
[20] M. Zhang, Q. Zhang, W. Liu, B. Zhou, "A policy-based reinforcement learning algorithm for intelligent train control," Journal of the China Railway Society, vol. 42, no. 01, pp. 69-75, 2020.
[21] M. Zhang, Q. Zhang, Z. Zhang, "A study on energy-saving optimization for high-speed railways train based on Q-learning algorithm," Railway Transport and Economy, vol. 41, no. 12, pp. 111-117, 2019.
[22] X. Wu and Z. Jin, "Research on train energy saving control strategy based on DDPG algorithm," Journal of Railway Science and Engineering, vol. 20, no. 02, pp. 483-493, 2023.
[23] M. Zhou, H. Dong, X. Zhou, W. Xu, L. Ning, "Reinforcement learning-based optimization of speed profile for high-speed train with temporary speed restriction," Journal of the China Railway Society, vol. 45, no. 02, pp. 84-92, 2023.
[24] S. Luo, J. Wei, X. Liu, L. Pan, "Collaborative deep reinforcement learning method for multi-objective parameter tuning," Transactions of Beijing Institute of Technology, vol. 42, no. 09, pp. 969-975, 2022.
[25] S. Su, Q. Zhu, Q. Wei, T. Tang, J. Yin, "A DQN-based approach for energy-efficient train driving control," Chinese Journal of Intelligent Science and Technology, vol. 2, no. 04, pp. 372-384, 2020.
[26] Q. Li, W. Li, Y. Cao, Y. He, "Research on energy-saving operation control method of urban rail train based on DDDQN," Journal of Railway Science and Engineering, pp. 1-12, 2024. [Online]. Available: https://doi.org/10.19713/j.cnki.43-1423/u.T20240343 (Accessed on: Oct. 13, 2024). DOI: 10.19713/j.cnki.43-1423/u.T20240343.

Xikai Liu was born in Gansu Province, China in 2001. He obtained his bachelor's degree in Traffic and Transportation from Beijing Jiaotong University, Beijing, China, in the year 2022 and is currently pursuing his master degree in Traffic and Transportation in Lanzhou Jiaotong University. His main research interests are transport system management and optimization.