Discover Applied Sciences

Research

A train trajectory optimization method based on the safety reinforcement learning with a relaxed dynamic reward

Ligang Cheng[1,2] · Jie Cao[1] · Xiaofeng Yang[3] · Wenxian Wang[2] · Zijian Zhou[3]

Received: 20 May 2024 / Accepted: 16 August 2024
Published online: 30 August 2024

Abstract
Train trajectory optimization (TTO) is an effective way to address energy consumption in rail transit. Reinforcement learning (RL), an excellent optimization method, has been used to solve TTO problems. Although traditional RL algorithms use penalty functions to restrict the random exploration behavior of agents, they cannot fully guarantee the safety of the process and results. This paper proposes a proximal policy optimization based safety reinforcement learning framework (S-PPO) for the train trajectory optimization, including a safe action rechoosing mechanism (SARM) and a relaxed dynamic reward mechanism (RDRM) combining a relaxed sparse reward and a dynamic dense reward. SARM guarantees that the new states generated by the agent consistently adhere to the environmental security constraints, thereby enhancing sampling efficiency and facilitating algorithm convergence. RDRM makes it easier for agents to obtain successful samples by relaxing time constraints, which also offers a better balance between exploration and exploitation. The experimental results show that S-PPO can significantly improve performance and obtain better train operation trajectories than soft constraint methods, and the convergence process is smoother. Finally, it was demonstrated that S-PPO exhibits good adaptability across various speed limit tracks.

Article Highlights

1. Discretize the train operation process based on distance and construct a Markov decision process model.
2. A safety reinforcement learning framework based on PPO is proposed to maintain the learning process within the constraints of boundaries.
3. A relaxed sparse reward which relaxes the constraint of train planned trip time is proposed to enhance the likelihood of agents completing tasks.
4. A dynamic dense reward can balance the contributions of time and energy consumption and offer enhanced feedback.

Keywords Energy efficient · Reinforcement learning (RL) · Train trajectory optimization (TTO) · Proximal policy optimization (PPO)

✉ Ligang Cheng, chenglg1110@163.com; Jie Cao, caoj@lut.edu.cn; Xiaofeng Yang, yangxiaofeng@crscd.com.cn; Wenxian Wang, wwx530@163.com; Zijian Zhou, zhouzijian@crscd.com.cn | [1]School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China. [2]School of Rail Transportation, Wuyi University, Jiangmen 529020, China. [3]CRSC Research & Design Institute Group Co., Ltd., Beijing 100000, China.

1 Introduction

Urban rail transit, as an efficient, safe, comfortable, and fast mode of transportation, has undergone significant development in the past few decades. However, along with this massive transportation capacity, there is inevitably an enormous demand for energy. Taking the Guangzhou Metro in China as an example, the total number of passengers transported in 2022 reached 2.358 billion, and the total operating energy consumption for the year was 1.882×10^9 (kw h), with train traction energy consumption accounting for 55.6%, totaling 1.047×10^9 (kw h) [1]. Energy-efficient train operation is a very effective measure for achieving energy conservation, emission reduction, and green transportation [2].

The keys to energy-efficient train operation include determining the speed profile of trains with minimal energy consumption and maintaining a timetable that can meet constraints such as train characteristics, track gradients, curves, and speed limits; this approach is also known as train trajectory optimization (TTO). The optimal train trajectory can describe the movement on the track and can be used as a basis for guiding train drivers during operation or as an input to automatic train operation (ATO).

To solve the TTO problem, many scholars have conducted extensive research on train operation control strategies and control methods, which are mainly divided into five categories: Pontryagin's maximum principle (PMP) [3–7], quadratic programming (QP) [8, 9], heuristic method [10–13], dynamic programming (DP) [14, 15] and reinforcement learning (RL) [16, 17]. The PMP faces significant challenges in TTO problems, especially when dealing with hard constraints on non-flat and multi speed-limited tracks, it is difficult to find the optimal conversion conditions. Kouzoupis et al. [18] used a multiple shooting method to transform the TTO into a nonlinear programming problem, and directly solved it using CasAdi (an open-source tool for numeric optimization implementing automatic differentiation in forward and reverse modes on sparse matrix-valued computational graphs) and IPOPT (a software for solving large-scale nonlinear optimization problems). And Wang and Goverde [8] solved it using the pseudospectral method. To improve the accuracy of the solution, the discrete scale of the TTO problem can be expanded, and the QP method requires a huge amount of computation and storage space to handle this large-scale problem. The heuristic methods often require considerable time to make control decisions and sometimes even lead to violent fluctuations in the speed profile that do not comply with the constraint. They regarded the train operation process as a multi-stage decision process [14, 15], and used the Bellman optimal equation and backpropagation method to obtain the optimal control strategy through iterative solving. To overcome the curse of dimensionality, approximate dynamic programming (ADP) is adopted to solve the TTO problem, and multiple value function approximation methods are designed to estimate the optimal value function, such as the rolling algorithm, the interpolation method, and neural networks [19–21].

The RLs allow agents to learn how to complete tasks through interaction with the environment, but they also meet many challenges, such as sparse rewards, balance between exploration and exploitation, sampling efficiency. To solve these problems, many classic RL algorithms have been proposed, such as Q-learning [22], deep Q-network (DQN) [23], advanced actor–critic algorithm (A2C) [24], deep deterministic policy gradient (DDPG) [25], proximal policy optimization (PPO) [26]. Liu et al. [16] proposed an intelligent control method based on the deep Q-network (DQN) to solve the TTO problem of heavy-haul trains. Liang et al. [27] used the asynchronous advanced actor–critic (A3C) to optimize the train speed profile and proposed a parameter update method with a weighted average of advantage values to address the convergence oscillation and degradation problem of the A3C. The train operation process can also be seen as a continuous control task, which is solved using DDPG [28, 29]. Pang et al. [30] addressed the problem of train trajectory reconstruction under interruption conditions, using the PPO model to consider train operation constraints and minimize total train delay, and proposed a train trajectory reconstruction scheme.

During the learning process, traditional RLs adopt a soft constraint approach, which involves setting a penalty function that matches the constraint to prevent agents from crossing boundaries and reaching unsafe states. However, in fields with complex transition dynamics and high-dimensional state-action spaces, this trial-and-error process may cause damage to the learning system when executing selected actions in certain states, affecting the efficiency of algorithm search.

To address this issue, safe reinforcement learning (SRL) has been proposed with the aim of satisfying given safety constraints and ensuring good system performance [31]. Several researchers have applied a Gaussian process to model safety constraints, which enables the algorithm to evaluate the safety of state-action pairs before accessing them to support safe learning [32–34]. The concept of *shielding* was first proposed by Alshiekh et al.[35]. During

learning, when an agent discovers that the current action is unsafe, it triggers *shielding* and uses an alternative action to cover the current action to ensure safety. Jeddi et al. [36] proposed a memory-augmented Lyapunov-based SRL model that enables agents to always meet the safety constraints of the environment. Zhou et al. [37] adopted a simplified system model to establish an SRL framework and effectively learned low-dimensional representations of safe regions through data-driven methods to obtain more accurate safe estimates, which expanded the applicability of the SRL framework.

In addition, RLs require a reward function to provide feedback [38], and learning performance largely depends on the design of the reward function [39]. Generally, reward functions are divided into two types: sparse rewards and dense rewards. The sparse reward function provides reward feedback only when completing tasks; therefore, it has strong anti-interference ability and can be made consistent with the task objectives [40–42]. When there are enough successful samples to provide reward feedback, the agent can learn the global optimal strategy [43]. However, in the early stage, its efficiency is relatively low. If the task is not completed, the agent will only receive samples with the same penalty reward, which makes it difficult for the algorithm to learn good strategies from these bad data. A dense reward function usually provides specific feedback for each state of the agent in a timely manner to distinguish different actions [44], which can maintain the continuity of learning and quickly guide the agent to approach high-value states. However, when designing a dense reward function, it is necessary to fully consider the possible interference from noise, as it is susceptible to interference from noise signals that may propagate and amplify through the Bellman equation [45, 46].

In summary, the PMP has limitations when dealing with TTO problems with hard constraints such as track slope and speed limitations. The RLs can obtain rewards through the constant interaction between agents and the environment to guide the continuous evolution of the algorithm. Therefore, it can well adapt to complex environmental constraints and has good generalization ability. The PPO algorithm, a typical RLs, adopt the mode of limiting strategy update amplitude, which enables it to maintain a high sample efficiency while effectively improving the stability and efficiency of training. Then, the SRL can effectively improve the soft constraint, which punishes state-action pairs that exceed the constraint, resulting in low sampling efficiency, a slow learning speed, and even breaking the constraints and obtaining the train running track beyond the safety limit.

Therefore, this paper proposes a PPO based safety reinforcement learning framework (S-PPO) for the train trajectory optimization, including a safe action rechoosing mechanism (SARM) and a relaxed dynamic reward mechanism (RDRM) combining a relaxed sparse reward and a dynamic dense reward. The SARM is proposed to guarantee both the safety of the learning process and the final result. The state transition process of the agent is evaluated by environmental knowledge, and when it is found that the agent's behavior exceeds the safety constraints, a new action is reselected to ensure that the next state reached by the agent always meets safety constraints, effectively improving the sampling efficiency. Notably, the SARM may be triggered at the beginning or middle of a state transition. The RDRM is designed to balance the potential convergence stability issues that the SARM may bring. The relaxed sparse rewards are obtained through extended planned trip time constraints, which makes it easier for the learning system to obtain samples that meet these constraints, greatly reducing the risk of the algorithm falling into local optima. The dynamic dense reward is a dynamic balance coefficient based on the initial velocity of the state, and is used to balance the contribution of running time and energy consumption to obtaining rewards in different states.

The remainder of this paper is organized as follows. In Sect. 2, the train operation model and the Markov decision process model of the train operation are formulated. In Sect. 3, we propose the S-PPO with the SARM and the RDRM for the TTO. In Sect. 4, simulations based on train and track data between Jiugong Station and Yizhuangqiao Station on the Beijing Metro Yizhuang Line verify the effectiveness of the proposed SVRDE and the energy efficiency of the proposed algorithm. In Sect. 5, conclusions are given.

2 Model construction

2.1 Basic train operation model construction

When studying the optimal operation of trains, a single-particle model [4, 47, 48] is always used to construct the kinematic system of trains. Assume that a train moves from the starting point $x = 0$ to the endpoint $x = X$. The running time $t = t(x) \in [0, T]$ and speed $v = v(x) \in [0, V]$ are used as the dependent variables of the model. Then, the train operation model can be expressed as follows:

$$\begin{cases} \dfrac{\mathrm{d}t}{\mathrm{d}x} = \dfrac{1}{v} \\ \dfrac{\mathrm{d}v}{\mathrm{d}x} = \dfrac{1}{M}\dfrac{\alpha_f f(v) - \alpha_b b(v) - w_0(v) - w_i(x)}{v} \end{cases} \tag{1}$$

where M is the mass of the train. $f(v) > 0$ and $b(v) > 0$ represent the maximum traction force and maximum braking force, respectively. $w_0(v) > 0$ expresses the resistance produced by friction and $w_i(x)$ is the resistance generated by gradients. α_f and α_b are the coefficients of traction and braking force utilization, respectively, must satisfy the constraint:

$$\begin{cases} 0 \le \alpha_f \le 1 \\ 0 \le \alpha_b \le 1 \end{cases} \tag{2}$$

It is worth noting that the single-particle model ignores the length of the train. When the train passes the gradient transformation points, the model cannot accurately express the force process, and there will be some bias in the description of the train operation. The size of the bias is related to $w_i(x)$ and the train length. Therefore, the single-particle model is generally established with the train center point as the reference point, which effectively reduces this bias and keeps its impact on the train within a tolerable range. Meanwhile, to transform TTO into an Markov decision process (MDP), this paper also ignores the impact of this bias.

The energy consumption E is an important indicator for measuring train operation trajectory control and can be expressed as:

$$E = \int_0^X \alpha_f f(v)\mathrm{d}x \tag{3}$$

The Hamiltonian function can be defined as:

$$H = -\alpha_f f(v) + \frac{\lambda_1}{v} + \frac{\lambda_2[\alpha_f f(v) - \alpha_b b(v) - w_0(v) - w_i(x)]}{v} \tag{4}$$

The Lagrangian function is represented as:

$$\mathrm{La} = H + \rho_1\alpha_f + \rho_2(1-\alpha_f) + \rho_3\alpha_b + \rho_4(1-\alpha_b) + \rho_5(V-v) \tag{5}$$

where $\rho_1 \ge 0, \rho_2 \ge 0, \rho_3 \ge 0, \rho_4 \ge 0, \rho_5 \ge 0$ are all Lagrangian multipliers and the adjoint variables must satisfy the following:

$$\begin{cases} \dfrac{\mathrm{d}\lambda_1}{\mathrm{d}x} = -\dfrac{\partial La}{\partial t} = 0 \\ \dfrac{\mathrm{d}\lambda_2}{\mathrm{d}x} = -\dfrac{\partial La}{\partial v} \end{cases} \tag{6}$$

According to the Karush–Kuhn–Tucker (KKT) condition, it can be concluded that

$$\frac{\partial \mathrm{La}}{\partial \alpha_f} = \left(\frac{\lambda_2}{v} - 1\right) f(v) + \rho_1 - \rho_2 = 0 \tag{7}$$

$$\frac{\partial \mathrm{La}}{\partial \alpha_b} = -\frac{\lambda_2}{v} b(v) + \rho_3 - \rho_4 = 0 \tag{8}$$

$$\frac{\partial \mathrm{La}}{\partial v} = -\alpha_f f'(v) - \frac{\lambda_1}{v^2} + \frac{\lambda_2}{v}(\alpha_f f'(v) - \alpha_b b'(v) - w_0'(v)) - \frac{\lambda_2}{v^2}(\alpha_f f(v) - \alpha_b b(v) - w_0(v) - w_i(x)) - \rho_5 = 0 \tag{9}$$

The complementary relaxation conditions are as follows:

$$\rho_1\alpha_f = \rho_2(1-\alpha_f) = \rho_3\alpha_b = \rho_4(1-\alpha_b) = \rho_5(V-v) = 0 \tag{10}$$

It can be seen that λ_2 has two critical values: $\lambda_2 = v$, $\lambda_2 = 0$. According to Pontryagin's maximum principle, five different conditions need to be considered to determine the control laws α_f and α_b such that the Hamiltonian function can reach the maximum value in the feasible region.

Condition 1: If $\lambda_2 > v$, since $f(v) > 0$, according to Eq. (7), $\rho_1 < \rho_2$. According to (10), due to $\rho_1, \rho_2 \geq 0$, $\rho_1 = 0$ and $\rho_2 > 0$ can be obtained; then, $\alpha_f = 1$. Similarly, according to (8), $\rho_3 - \rho_4 > 0$ can be obtained; since $\rho_3\alpha_b = \rho_4(1 - \alpha_b) = 0$, $\alpha_b = 0$. The train operates with maximum traction force (*MT*).

Condition 2: If $\lambda_2 = v$, according to Eq. (8), it is easy to find that $\rho_3 - \rho_4 = b(v) \geq 0$; and according to Eq. (10), $0 \leq \alpha_f \leq 1$ and $\alpha_b = 0$.

When $\lambda_2 = v$, we can obtain equation $\frac{d\lambda_2}{dx} = \frac{dv}{dx}$ and substitute Eqs. (1), (6) and (9) into it and simplify them:

$$(w_0'(v)+\rho_5)v^2+\lambda_1 = 0 \tag{11}$$

In this condition, the speed is held at a certain value $v = v_c$ [v_c is the positive solution of Eq. (11)]. This indicates that the train operates at a constant speed with only partial traction force, which is called traction cruising (*CR-T*).

Condition 3: If $0 < \lambda_2 < v$, it can be inferred that $\rho_1 > \rho_2$ and $\rho_3 > \rho_4$ according to (7) and (8), respectively. Then, according to (10), we can obtain $\alpha_f = \alpha_b = 0$. Under these conditions, both the traction and braking forces of the train are zero, which is called coasting (*CO*).

Condition 4: If $\lambda_2 = 0$, similar to condition 2, we can obtain $\rho_1 > 0$ and $\rho_3 = \rho_4 = 0$, corresponding to $\alpha_f = 0$ and $0 \leq \alpha_b \leq 1$.

When $\lambda_2 = 0$, we can construct equations equation $\frac{\lambda_2}{v} = 0$ and substitute Eq. (9):

$$\rho_5 v^2-\lambda_1 = 0 \tag{12}$$

If $\rho_5 = 0$, then Eq. (12) does not hold. If $\rho_5 > 0$, according to Eq. (10), $v = V$, Eq. (12) may hold. The speed is held at a certain value $v = V = \sqrt{\lambda_1/\rho_5}$. The train operates at a constant speed with only partial braking force; this process is called braking cruising (*CR-B*).

Condition 5: If $\lambda_2 < 0$, as above, we can obtain $\rho_1 > 0$ and $\rho_4 > 0$, so $\alpha_f = 0$ and $\alpha_b = 1$. The train operates with a maximum braking force (*MB*).

Under both *CR-T* and *CR-B* operating conditions, trains can maintain a constant speed; these conditions are collectively referred to as cruising (*CR*).

In summary, the optimal train operation strategy consists of a sequence of four operation regimes: *MT*, *CR*, *CO* and *MB*. Therefore, The continuous train operation is simplified into these four actions: *MT*, *CR*, *CO* and *MB*, which can greatly reduce the dimension of the action space and make TTO easier to solve.

2.2 The Markov decision process of train operation

The MDP is a classical sequential decision process and can be defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ represents the state transfer function, $\mathcal{R}$ is the reward function, and γ is the discount factor.

The agent and environment need to constantly interact, and the current action used in the interaction not only affects the immediate reward but also influences the subsequent state through future rewards. In step i, the agent's state is to interact with the environment through actions, reach a new state, and receive a reward. At each step i, an agent in state $s_i \in \mathcal{S}$ interacts through sampling action $a_i \in \mathcal{A}$ and transitions to a new state s_{i+1}, receiving reward r_i; the process of this state transition is denoted as $s_i \xrightarrow[r_i]{a_i} s_{i+1}$.

Here, the TTO problem is transformed into an MDP by the discretization. Common discretization methods include time discretization [49, 50], distance and velocity discretization [48], and distance discretization [18, 51].

The process of track discretization is shown in Fig. 1. Based on the gradient and the inflection point of the speed limit of the track, the track is discretized into H large sections (each with a length of l_h, $(h = 1, 2, \cdots, H)$). To ensure accuracy, each large section needs to be further discretized. The maximum discretization step size of distance is Δl, and each large segment can be discretized into $m_h = \text{ceil}(l_h/\Delta l)$ small segments. The dimension of the discrete TTO problem is $N = \sum_{h=1}^{H} m_h$.

The advantage of this discretization method is that it can ensure fixed properties (speed limit, slope and –if considered –curvature) for each small interval, and can solve the problems of unstable force and speed constraints on trains.

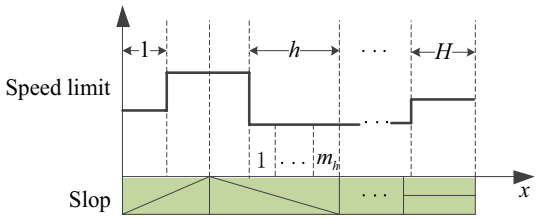

Fig. 1 The principle of track discretization

To consider the adaptability of the algorithm to temporary speed limits, this paper also takes adjacent temporary speed limit information in front of the train as important state information when defining the state of the agent. Assume that the number of temporary speed limit intervals is Num_{sl}; the positions of the temporary speed limit intervals are $[xsl_j^{start}, xsl_j^{end}], j = 1, 2, \cdots Num_{sl}$; and the speed limit value is vsl_j. Then, the state is defined as follows:

$$s_i = (x_i, v_i, \Delta xsl_i, \Delta vsl_i, T_i, E_i), s_i \in \mathcal{S} \tag{13}$$

where the speed difference from the temporary speed limit is $\Delta vsl_i = vsl_j - v_i$.

The distance from the current position to the start of the next adjacent temporary speed limit interval is represented as follows:

$$\Delta xsl_i^{start} = \begin{cases} xsl^{start} - x_i & x_i < xsl_{Num_{sl}}^{start} \\ 0 & x_i \geq xsl_{Num_{sl}}^{start} \end{cases} \tag{14}$$

T_i and E_i represent the running time and energy consumption, respectively, which can be calculated as follows:

$$\begin{cases} T_i = \sum_{k=1}^{i} \int_{x_{k-1}}^{x_k} \frac{1}{v(x)} \mathrm{d}x \\ E_i = \sum_{k=1}^{i} \int_{x_{k-1}}^{x_k} \alpha_f f(v) \mathrm{d}x \end{cases} \quad k = 1, \cdots, i \text{ and } i = 1, 2, \cdots N \tag{15}$$

In the interval $[0, X]$, the total trip time of the train is $T_N \in [T_{\min}, T_{\max}]$, and the total energy consumption is $E_N \in [E_{\min}, E_{\max}]$.

Notably, for the fixed-speed limit sections, the speed limit is lower than the maximum speed limit, and it has the same spatial distribution characteristics as the temporary speed limit during train operation tasks; therefore, this method is also applicable to these sections.

Based on Sect. 2.1, the action space can be defined as:

$$\mathcal{A} = \{MT, CR, CO, MB\} \tag{16}$$

The deviation of trip time and energy consumption are important indicators for measuring the trajectory of train operation. Therefore, the objective function is defined as follows:

$$\arg\min_{\pi^*} \ J(\pi) = \{J_{\Delta T}(\pi), J_E(\pi)\} \tag{17}$$

where π is the train operation strategies; $J_E = E_N$, $J_{\Delta T} = |T_N - T_P|$, and T_P is the planned trip time.

The ideal total trip time should be consistent with the planned trip time, but in actual train control scenarios, there may be some deviation Δt between them, and small Δt is allowed. By simplifying the time deviation objective as a constraint:

$$|T_N(\pi) - T_P| \leq \Delta t \tag{18}$$

The TTO problem can be transformed into a single-objective optimization problem, and objective function is:

$$\arg\min_{\pi^*} \ J(\pi) = J_E(\pi) \tag{19}$$

3 The safety reinforcement learning framework

In this section, the PPO algorithm is introduced, and a safe action rechoosing mechanism and a relaxed dynamic reward function are proposed to improve the performance of the algorithm.

3.1 The PPO algorithm

PPO algorithm is a classic RL algorithm consisting of two networks, the actor network and the critic network, which can be defined by weights $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. The actor network generates the probability distribution of possible actions, which is used to choose the best action. The critic network assesses the value of the current state and guides the network of actors to make better decisions. The PPO optimizes a *clipped* surrogate objective function using mini-batch stochastic gradient ascent, which is given by

$$L(\theta) = \hat{\mathbb{E}}\left[\min\left(\rho_i(\theta)A^{\pi_{\theta_{\text{old}}}}\left(s_i, a_i\right), clip\left(\rho_i(\theta), 1-\varepsilon, 1+\varepsilon\right)A^{\pi_{\theta_{\text{old}}}}\left(s_i, a_i\right)\right)\right] \tag{20}$$

$\rho_i(\theta) = \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{\text{old}}}(a_i|s_i)}$ denotes the probability ratio between the previous and updated policies. ε is the hyperparameter, represents the range for truncation. $A^{\pi_{\theta_{\text{old}}}}(s_i,a_i)$ is an advantage function estimated by the generalized advantage estimation

$$A^{\pi_{\theta_{\text{old}}}}(\boldsymbol{s}, \boldsymbol{a}) = \delta_i + (\gamma\lambda)\delta_{i+1} + \cdots + (\gamma\lambda)^{U-i}\delta_U \tag{21}$$

where $\delta_i = r_i + \gamma V_\omega(s_{i+1}) - V_\omega(s_i)$.$V_\omega(s_i)$ is the value of state s_i and r_i is the reward at i time step. U denotes the size of mini-batch. γ and λ are discount factor and GAE parameter, respectively. clip($\cdot$) is clip function that can prevent the disastrous performance loss caused by the high variance inherent in the strategy gradient method by conservatively optimizing the strategy. If $A^{\pi_{\theta_{\text{old}}}}(s_i,a_i) > 0$, $\rho_i(\boldsymbol{\theta})$ is cliped at $1+\varepsilon$; On the contrary, if $A^{\pi_{\theta_{\text{old}}}}(s_i,a_i) < 0$, then $\rho_i(\boldsymbol{\theta})$ is cliped at $1-\varepsilon$.

$$L(\omega) = \hat{\mathbb{E}}\left[\left(V_\omega(s_i) - \hat{V}_i\right)^2\right], \quad \text{where } \hat{V}_i = \sum_{j=i}^{U}\gamma^{j-i}r_j \tag{22}$$

At the end, the critic network and actor network are updated by:

$$\boldsymbol{\omega}_{\text{new}} \leftarrow \boldsymbol{\omega}_{\text{now}} - \alpha_c \cdot \nabla_\omega L(\boldsymbol{\omega}) \tag{23}$$

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{now}} + \alpha_a \cdot \nabla_\theta L(\boldsymbol{\theta}) \tag{24}$$

3.2 Safe action rechoosing mechanism

In practical applications, the environment is often bounded. To ensure that each state transition satisfies the constraints of the environmental boundary, traditional RLs punish the state–action pairs that exceed the environmental limit and keep the state unchanged, i.e., $s_i \xrightarrow[r_i]{a_i} s_i$, to correct unsafe states. This approach is called an unsafe state maintenance mechanism. This method of collecting data through trial and error to obtain the optimal strategy can incur significant costs and may even cause damage to the learning system during application. Therefore, this paper presents an S-PPO algorithm with SARM and RDRM based on the PPO by analyzing the characteristics of the TTO problem. The unsafe actions discovered during interactions with the environment are rechosen to improve the algorithm learning efficiency and stability. To ensure the safety of train operation, safety law must be met:

$$v(x) < V_{\max}(x), x \in [x_i, x_{i+1}] \tag{25}$$

Where $V_{\max}(x)$ is the actual speed limit curve generated by the automatic train protection system (ATP).

However, in reality, when the train speed approaches the speed limit, an inappropriate action may violate a safety law. There are two specific situations to describe:

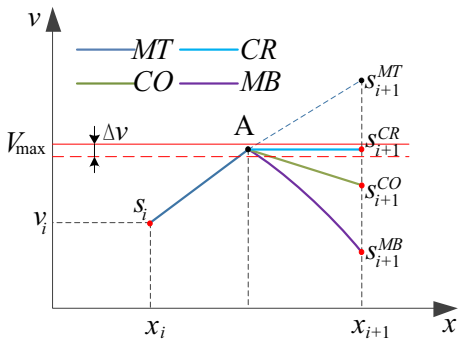

Fig. 2 Rechoosing plan for a constant speedlimit interval

Situation 1: In the constant speed limit interval, as shown in Fig. 2, when the train's speed $v(x_i)$ approaches the speed limit $V_{\max}(x_i)$ in a state s_i and action $a_i = \{MT\}$, e.g., $s_i \xrightarrow{MT} s_{i+1}^{MT}$, then $v^{MT}(x_{i+1}) > V_{\max}(x_{i+1})$, which violates safety law (25). However, if the action can be rechosen as $a_i \in \{CR, CO, MB\}$ at point A ($V_{\max}(x_A) - \Delta v \le v(x_A) < V_{\max}(x_A)$, where Δv is the allowable error for approaching the speed limit), then the corresponding state transitions to $s_i \xrightarrow{CR,CO,MB} s_{i+1}^{MT}, s_{i+1}^{CO}, s_{i+1}^{MB}$ and the speed meets condition (25), which can ensure the safety of the train.

Situation 2: In the braking speed limit interval, as shown in Fig. 3, when the train's speed $v(x_i)$ approaches the speed limit $V_{\max}(x_i)$ in a state s_i and action $a_i \in \{MT, CR, CO\}$, if the speed remains constant within the interval and transitions to the next state $s_{i+1} \in \{s_{i+1}^{MT}, s_{i+1}^{CR}, s_{i+1}^{CO}\}$, the corresponding speed $v^* > V_{\max}(x_{i+1}), v^* \in \{v^{MT}, v^{CR}, v^{CO}\}$, does not comply with safety law (25). If the actions at points A, B, and C can be reset, it can ensure that $s_{i+1} = \{s_{i+1}^{MB}\}$ meets the requirements of (25).

Specifically, in S-PPO, as shown in Fig. 4, the probability distribution $Pdist_{\mathcal{A}}$ of each action in action space $\mathcal{A}$ can be generated based on the policy network $\pi(\cdot|s_i;\boldsymbol{\theta}_{now})$. The action a_i sampled based on the probability distribution $Pdist_{\mathcal{A}}$

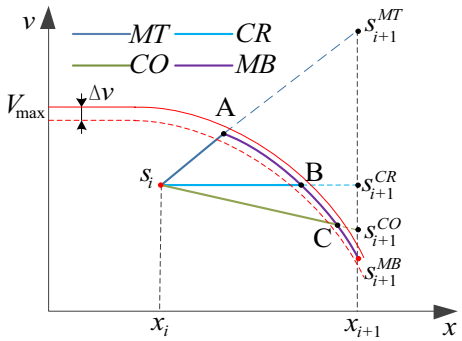

Fig. 3 Rechoosing plan for a braking speed limit interval

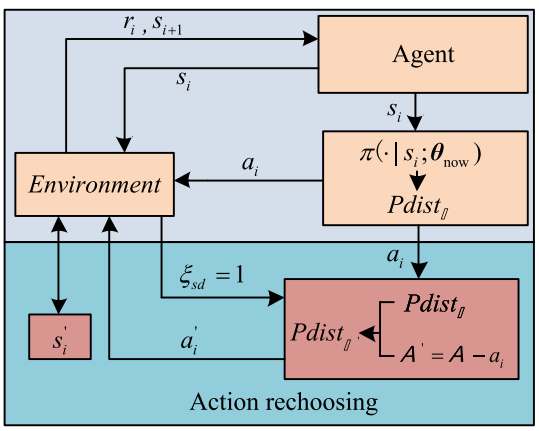

Fig. 4 The safe action rechoosing mechanism

is used to interact with the environment. Environmental knowledge is used to evaluate the safety of state–action pairs, and the safety judgment coefficient is ξ_{sd}.

$$\xi_{sd} = \begin{cases} 0 & v(x) \le V_{\max}(x) \\ 1 & v(x) > V_{\max}(x) \end{cases} \tag{26}$$

If $\xi_{sd} = 1$, continuing to take the action a_i will cause the train to exceed the speed limit, which does not satisfy the safety law. Therefore, the SARM will be triggered. To prevent the new action a_i' after rechoosing from being consistent with the original action a_i, it is necessary to remove unsafe actions and reconstruct the action space $\mathcal{A}'$ ($\mathcal{A}' = \mathcal{A} - a_i$). Then, the probability distribution of the remaining actions is normalized to obtain the reconstructed probability distribution $Pdist_{\mathcal{A}'}$, which is used to obtain a new action a_i'. Therefore, when performing the SARM, new actions are sampled based on $Pdist_{\mathcal{A}'}$. The actions with higher adoption probabilities can have more opportunities to be reselected. The pseudocode of the safe action rechoosing mechanism is shown in algorithm 1.

Algorithm 1 SARM

1: Input s_i and a_i to the *Environment*.
2: **if** $\xi_{sd} = 1$ **do**
3: Save $s_i^{'}$ from the *Environment*.
4: Delete unsafe actions $A^{'} = A - a_i$.
5: Normalization $Pdist_{A^{'}}$.
6: Obtain $a_i^{'}$ by sampling.
7: Input $s_i^{'}$ and $a_i^{'}$ to the *Environment*.
8: **end if**

3.3 Relaxed dynamic reward function construction

The performance of RL agent's learning largely depended on the reward function design. In [16], a dense reward strategy has been designed, which punishes unsafe operations and encouraging the release of air brakes under specific conditions through a positive reward. [30] constructs a sparse reward, which the system gives a large immediate reward to agent when the train reaches the final position, otherwise the instantaneous reward obtained by the system agent is always 0. Lin et al. [52] and Haung et al. [53] adopt a weighted average approach to incorporate multiple objectives into the reward function.

This paper combines the advantages of sparse rewards and dense rewards to design a reward function, called the RDRM, which includes the relaxed sparse reward and the dynamic dense reward. The planned trip time is limited to a narrow range, which makes it difficult for the algorithm to collect samples that meet the constraints during learning. The relaxed sparse reward function is established to increase the probability of the agent completing tasks by relaxing the time constraints of the train operation plan, which can enable the agent to obtain more successful samples and accelerate the speed of learning the optimal strategy. Furthermore, the dynamic dense reward function is established based on the average planned speed of the train, which can balance the contributions of time rewards and energy consumption rewards according to the states of different trains and provide better feedback.

3.3.1 The relaxed sparse reward

In the iterative learning process, exploring successful strategies requires meeting time constraints $\left|T_N(\pi) - T_P\right| \le \Delta t$. The small Δt makes it difficult for the algorithm to obtain successful experience to train the learning network, seriously

affecting the learning efficiency of the network. Therefore, it is necessary to relax the time constraint to improve the probability of successful strategy acquisition during the algorithm exploration process.

According to this conclusion, if $T_N(\pi^*) = T_P$, we can propose several typical optimal strategies π_k^*, $k = 1, 2, 3, 4$ and $T_P^1 < T_P^2 < T_P^3 < T_P^4$. The trajectories are shown in Fig. 5 and Table 1. According to Eqs. (15), the train's energy consumption can be expressed as:

$$J_E = E_N = E_{MT} + E_{CR-T} = \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} f(v)\mathrm{d}x + \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} \alpha_f f(v)\mathrm{d}x = \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} vf(v)\mathrm{d}t + \sum_{j=1}^{n} \int_{t_{j-1}}^{t_j} \alpha_f vf(v)\mathrm{d}t \tag{27}$$

where E_{MT} and E_{CR-T} are the maximum traction energy consumption and traction cruise energy consumption, respectively; m and n represent the numbers of sections using maximum traction and traction cruising, respectively. The trip times for trains to maintain MT in different strategies are $T_{MT}^1 = T_{MT}^2 = T_{MT}^3 > T_{MT}^4$, and the energy consumption is $E_{MT}^1 = E_{MT}^2 = E_{MT}^3 > E_{MT}^4$. In the CR-T stage, $T_{CR-T}^1 > T_{CR-T}^2 > T_{CR-T}^3 = T_{CR-T}^4$ and $E_{CR-T}^1 > E_{CR-T}^2 > E_{CR-T}^3 = E_{CR-T}^4$. Therefore, it is easy to deduce $J_E^{\pi_1^*} > J_E^{\pi_2^*} > J_E^{\pi_3^*} > J_E^{\pi_4^*}$. Furthermore, due to the negative correlation between $T_N(\pi^*)$ and the average speed $\overline{v}_{\pi^*}$ of the strategy, when v is higher, it is positively correlated with w_0. The higher the speed is, the more energy the train needs to consume to overcome resistance.

As a result, for an optimal train operation strategy π^*, if $T_N(\pi^*) = T_P$, $T_P \in [T_{\min}, T_{\max}]$, then T_P is negatively correlated with the energy consumption of optimal strategy $E_N(\pi^*)$ for the TTO problem on a straight track. And we can draw the following:

$$\arg\min_{\pi^*} J(\pi) = \arg\min_{\pi^*} J_E(\pi), \quad T_N(\pi) \in [T_{\min}, T_P] \tag{28}$$

When $T_N^{\pi^*} \in [T_{\min}, T_P]$; $J_E^{\pi^*}(T_N^{\pi^*})$ monotonically decreases, i.e.,$\min(J_E^{\pi^*}) = J_E^{\pi^*}(T_P)$, $T_N^{\pi^*} \in (T_P, T_{\max}]$; thus, the Gaussian function is adopted. Therefore, by relaxing Eq. (18), a train trip-time reward function R_T is constructed, as shown in Fig. 6.

$$R_T = \begin{cases} 500 & T_N \in [T_{\min}, T_P] \\ 500e^{-(T_N - T_P)^2/20} & T_N \in (T_P, T_{\max}] \end{cases} \tag{29}$$

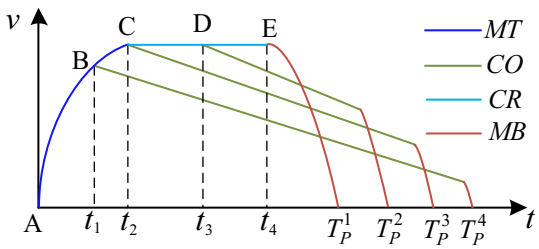

Fig. 5 Schematic diagram of the optimal strategy trajectories

Table 1 Correspondence between the optimal strategy and trajectory

Strategy	π_1^*	π_2^*	π_3^*	π_4^*
Trajectory	$ACET_P^1$	$ACDT_P^2$	ACT_P^3	ABT_P^4

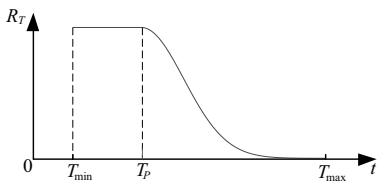

Fig. 6 The train trip reward function curve

The energy consumption reward function R_E reflects the energy consumption level of the train.

$$R_E = 500e^{-(E_N - E_{\min})/(E_{\max} - E_{\min})} \tag{30}$$

Therefore, the relaxed sparse reward function is represented as follows:

$$sparse\_r_i = \begin{cases} 0 & i < N \\ \beta R_T + (1-\beta)R_E & i = N \end{cases} \tag{31}$$

where β represents weighting factor for the trip time weight and energy consumption weight coefficients. An agent guided by sparse rewards can in principle have higher consistency with the task; thus, larger values of $sparse\_r_N$ is needed to highlight the contribution of sparse rewards. In the TTO, the constraint of planning travel time needs to be strictly followed, so we pay more attention to time rewards. Therefore, this paper takes $\beta = 0.6$.

3.3.2 The dynamic dense reward

A dense reward can enhance the exploration ability of algorithms [39]. Based on two objective of time and energy consumption, we design an average dense reward function *ave_r*.

$$ave\_r_i = \begin{cases} \beta r_t^i + (1-\beta) r_e^i & i < N \\ \beta R_T + (1-\beta)R_E & i = N \end{cases} \tag{32}$$

Here, the dense time reward function r_t should satisfy the planned travel time constraint.

$$r_t^i = e^{-(|T_i - T_P|)/T_P} \tag{33}$$

The dense energy consumption reward function r_e should guide agents to approach the optimal strategy for energy consumption as well as possible.

$$r_e^i = e^{-(|E_i - E_{\overline{v}_p}|)/E_{\overline{v}_p}} \tag{34}$$

where the planned average trip time is $\overline{v}_p = X/T_P$. $E_{\overline{v}_p}$ is the benchmark energy consumption, where the train accelerates to $\overline{v}_p$ with *MT*, then moves to *CR* and *MB*, adopts coasting and completes all generated energy consumption steps. Making the energy consumption appropriately smaller than $E_N(\pi^*)$ is more beneficial for the algorithm learning process. Notably, the dense reward function must have the ability to guide the agent toward the optimal goal.

The larger the value of r_t^i is, the more it helps the agent accelerate and save time. Similarly, the larger r_e^i is, the more it helps the agent slow down and reduce energy consumption. In the initial stage, increasing the contribution of r_t^i is more advantageous for agents to achieve higher speed. In contrast, in the final stage, when the speed $v \to 0$, increasing the contribution of r_e^i is more advantageous. Moreover, considering the constraints of planned travel time, a linear balance benchmark function $Fbal(x)$ is constructed based on the planned average trip time $\overline{v}_p$.

$$Fbal(x) = -\frac{20}{X}x + \frac{X}{T_P} + 5 \tag{35}$$

The dynamic balance coefficients ξ_e and ξ_t used to balance time rewards and energy consumption rewards.

$$\begin{cases} \xi_e = \dfrac{v_i - Fbal(x_i)}{V_{\max}(x_i)} + 0.5 \\ \xi_t = -\dfrac{v_i - Fbal(x_i)}{V_{\max}(x_i)} + 0.5 \end{cases} \tag{36}$$

The dynamic dense rewards are represented as follows:

$$dense\_r_i = \begin{bmatrix} \xi_e \\ \xi_t \end{bmatrix} \begin{bmatrix} r_e^i \\ r_t^i \end{bmatrix} \tag{37}$$

The SARM reschoosing actions beyond the boundary, the agent is inevitably able to transition from one state to the new state with each iteration, rather than remaining in the current state. In training, the agent completes a episode after N iterations. Therefore, when $i = N$, only more accurate sparse rewards are used to guide the learning process. The relaxed dynamic reward function is

$$Reward_i = \begin{cases} dense\_r_i & i < N \\ sparse\_r_i & i = N \end{cases} \tag{38}$$

The pseudocode of S-PPO is shown in Algorithm 2.

Algorithm 2 The pseudocode of S-PPO

Input: Necessary parameters, Max_epis, n_epochs, α_c, α_a, γ, λ, ε, U, β, Δl, M and T_P.

Output: Estimation of optimal strategy $\pi(\theta)$

Initialize the actor network parameters $\boldsymbol{\theta}$.
Initialize the critic network parameters $\boldsymbol{\omega}$.
Initialize the buffers $\mathcal{B} \neg \mathcal{A}$
Discrete track space and its dimension is N
for *episode* =1:*Max_epis* **do**
  Initialize the train state s
  for $i = 1, 2, \cdots N$ **do**
    Select a_i according to the policy $\pi(\cdot \mid s_i; \boldsymbol{\theta})$.
    Input s_i and a_i to the *Environment*.
    if $\xi_{sd} = 1$ **do** Algorithm 1 **end**.
    Receive new state s_{i+1}, $Reward_i$ and ***done*** from the *Environment*.
    Collect the trajectory $\mathcal{B} \neg \mathcal{B} \bigcup (s_i, a_i, Reward_i, s_{i+1})$
    if $|\mathcal{B}| = U$ **then**
      for j =1:*n_epochs* **do**
        Sample from the replay buffer $\mathcal{B}$
        Calculate $L(\theta)$ and $L(\omega)$ by (20) and (22), respectively.
        Update network parameters $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ according to (23) and (24), respectively.
      end for
    end if
  end for
end for

Table 2 Static speed limit data of the track

Sequence	Starting point (*m*)	Endpoint (*m*)	Speed limit (*km/h*)
1	0	130	54
2	130	1840	80
3	1840	1975	54

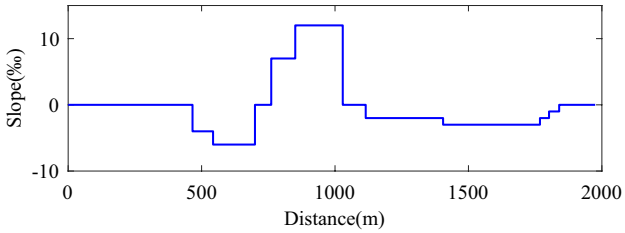

Fig. 7 The slope data of the track

Table 3 The values of the parameters

Name	Symbol	Value
Learning rate of critic network	α_c	0.0003
Learning rate of actor network	α_a	0.0003
Discount factor	γ	0.99
GAE parameter	λ	0.95
Clipping rate	ε	0.2
Maximum number of episode	*Max_epis*	5000
Number of epochs	*n_epochs*	5
Mini-batch size	*U*	40
Weighting factor	β	0.6
Maximum discretization step size	Δl	30 m
Planned trip time	T_P	130 s
Allowable time deviation	Δt	1 s
Mass of the train	*M*	$1.94*10^5$ kg

4 Experimental simulation and analysis

We implement the proposed S-PPO method via three fully connected hidden layers with 120 hidden units, and the numbers of output layers of the strategy network and the value network are one and four, respectively. This simulation is based on the line data from Jiugong Station to Yizhuangqiao Station on the Beijing Subway Yizhuang Line [28]. Table 2 shows the static speed limit data of Jiugong Station and Yizhuangqiao Station, and the slopes of the track are shown in Fig. 7. The proposed algorithm is implemented in Python on a computer with an Intel Core i7-10700 CPU @2.90 GHz and 32 GB RAM running Windows 10 × 64 Edition.

The parameters used in the algorithm are shown in Table 3. A smaller discretization step Δl can improve the accuracy of the results, but it also makes the calculation more complex. Therefore, this paper takes $\Delta l = 30$ (m), corresponding to which the track is discretized into $N = 73$ sub segments and takes $\Delta t = 1$ (s).

We follow the hyperparameters recommended by PPO [26] and set clipping rate $\varepsilon = 0.2$. And we set the discount factor to a higher value, $\gamma = 0.99$, which can encourage agents to focus more on long-term rewards. In addition, we set the mini-batch size to $U = 40$ and the Number of epochs $n\_epoch = 5$. As the PPO algorithm is insensitive to the change in the learning rate, we always keep learning rates of critic network and actor network the same and select 0.0001, 0.0003, and 0.0005 earning rates in the experiment. The experimental results are shown in Fig. 8. When learning rates is 0.0001, the convergence speed is relatively slow. The larger learning rate of 0.0005, although improving the convergence speed, also causes oscillations in the convergence curve in the later stages of iteration. Therefore, we choose a moderate learning rate of 0.0003 in this paper, which can achieve faster convergence speed and maintain a stable convergence curve.

Effectiveness Test: We compare the optimization effects of the S-PPO and the traditional PPO without safety protection measures on the TTO problem to verify the effectiveness of the SARM. A comparative experiment is conducted under three different reward functions, namely, the sparse reward function (31), the average dense reward function (32), and the relaxed dynamic reward function (38). The combination table numbers for the algorithms and rewards are shown in Table 4. Furthermore, we conducted a statistical analysis of the unsafe action counts of S-PPO and PPO with the relaxed dynamic reward (*Max_ep* = 5000) to further demonstrate the effectiveness of the safe action rechoose mechanism. In

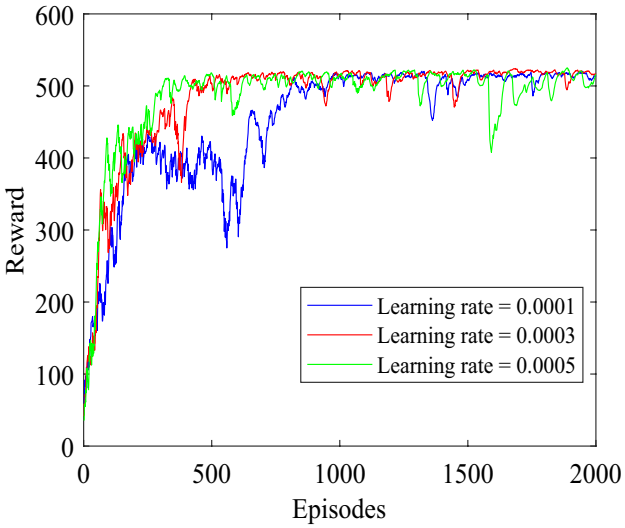

Fig. 8 The slope data of the track

Table 4 The combination table numbers for the algorithms and reward functions

Sequence	Reward function		Algorithm
①	Relaxed dynamic reward	(38)	S-PPO
②	Average dense reward	(32)	S-PPO
③	Sparse reward	(31)	S-PPO
④	Relaxed dynamic reward	(38)	PPO
⑤	Average dense reward	(32)	PPO
⑥	Sparse reward	(31)	PPO

addition, we compared the performance of S-PPO with two other excellent train operation methods (i.e. A3C [27], DQN [16]), as the train operation processes in these methods are all modeled as MDPs.

Universality Test: Generally, the most important factor affecting the train trajectory is the maximum speed constraint, which determines the basic shape of the train operation trajectory. In practical applications, trains often need to operate on speed limited tracks with different spatial characteristics. Therefore, to verify the performance of S-PPO on tracks with different speed limits, we randomly added speed limited sections with different characteristics on the original track to test its adaptability.

4.1 Effectiveness test

Due to the lack of a safety protection mechanism, the traditional PPO uses soft constraints to handle unsafe risks such as those exceeding environmental boundaries. Therefore, a penalty value needs to be set for such behavior, and in this experiment, the penalty value $r_{\text{penalty}} = -1$is chosen. The convergence curves and the train trajectories of the reward of the S-PPO and the PPO algorithms for three different rewards are shown in Figs. 9 and 10, respectively. Table 5 presents the results of the energy consumption, total trip time, and time deviation with respect to the planned trip time for the S-PPO and PPO algorithms for three different rewards.

1. Figure 9 shows that the three combinations ①, ②, and ③ of the S-PPO algorithm converge to better reward values than the corresponding three combinations ④, ⑤, and ⑥ of the PPO algorithm. And, ① and ② converge faster and smoother than ④ and ⑤, respectively. Although ③ has poorer convergence speed and stability than ⑥. It can jump out of local optima in the later stage of iteration, causing continuous oscillation, which requires more iterations (exceeding *Max_ep*) to reconverge to a stationary state and obtains a better train operation control model. The three combinations of ④, ⑤ and ⑥ all have significant oscillations in the early stages of training. This is because PPO

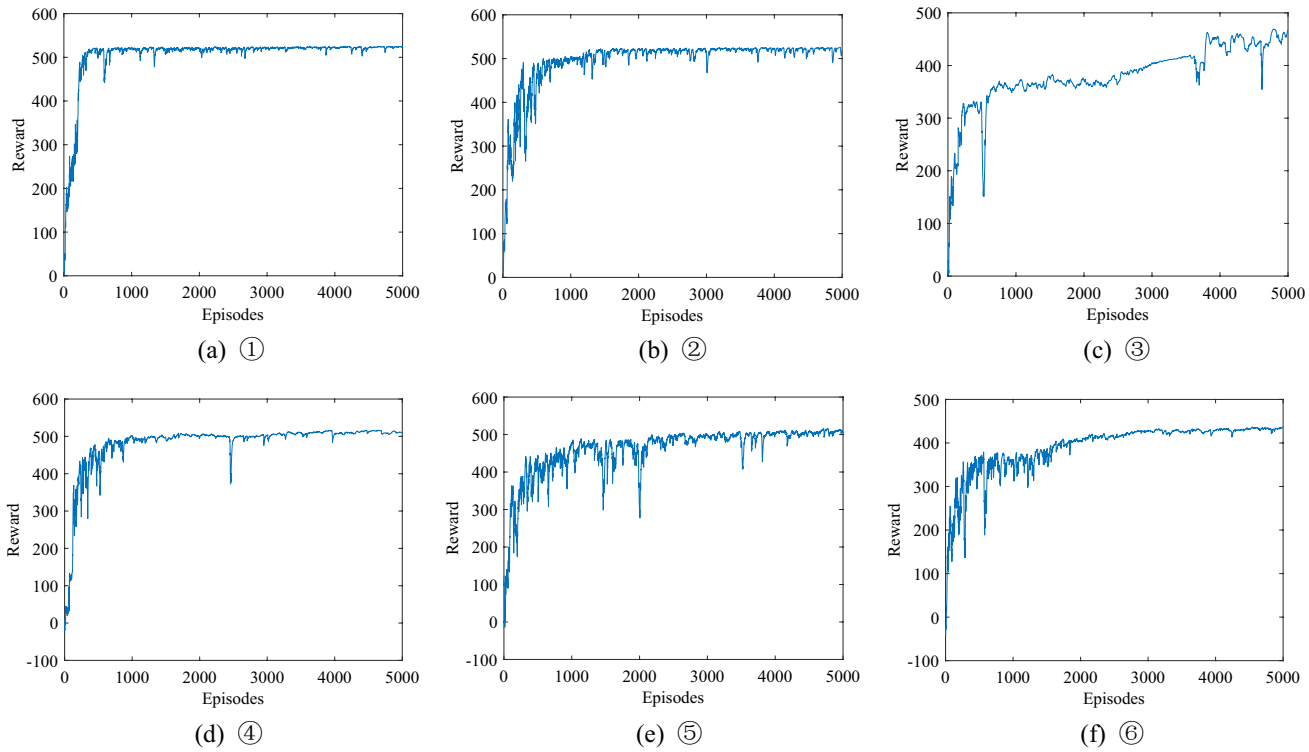

Fig. 9 The convergence curves S-PPO and PPO based on relaxed dynamic reward, average dense reward and sparse reward

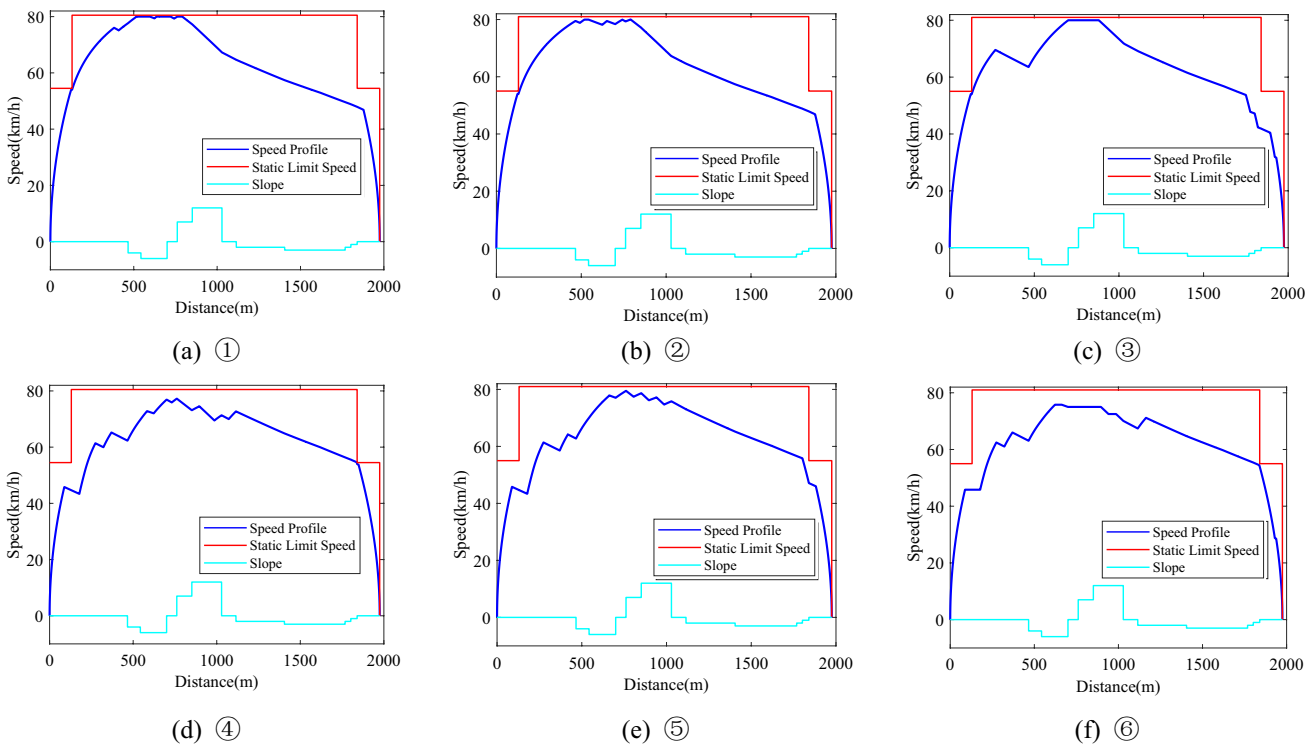

Fig. 10 The trajectories of train operating using S-PPO and PPO based on relaxed dynamic reward, average dense reward and sparse reward

Table 5 The results of the energy consumption E_N, the total trip time T_N, and the time deviation $J_{\Delta t}$ with respect to the planned trip time for the S-PPO and PPO algorithms for three different rewards

Algorithms	Relaxed dynamic reward			Average dense reward			Sparse reward		
	E_N	T_N	$J_{\Delta t}$	E_N	T_N	$J_{\Delta t}$	E_N	T_N	$J_{\Delta t}$
S-PPO	21.13	130.06	0.06	21.47	130.07	0.07	22.06	129.79	0.21
PPO	22.32	130.04	0.04	22.41	129.43	0.57	22.61	130.02	0.02
	5.63%	–	< 1 s	4.37%	–	< 1 s	2.49%	–	< 1 s

does not have a safe action protection mechanism. At the beginning of training, the learning system has not formed a relatively stable strategy model, making it easier for unsafe actions to occur near the environmental boundary, causing damage to the learning system. The three combinations of S-PPO with SARM ①, ②, and ③ can effectively avoid the disturbance of unsafe behavior during the training process, which enables the algorithm to better balance exploration and development and enhance the learning efficiency of the algorithm and the smoothness of the training process.

In addition, the convergence curve of ① and ④ is the fastest and smoothest, followed by ② and ⑤, while the convergence process of ③ and ⑥ is slower and the fluctuation is relatively larger. Therefore, compared with average dense reward and sparse reward, relaxed dynamic reward can better adjust the scale of feedback according to different state-action pairs information, enabling the algorithm to obtain sample information faster and more effectively and promote its learning process.

2. Figure 10 shows that the three trajectories of the S-PPO algorithm, ①, ②, and ③, have fewer operational changes and are smoother than the three trajectories of the PPO algorithm, ④, ⑤, and ⑥. ①, ②, and ③ are able to quickly increase the train speed with the *MT* in the early stage and then maintain the speed at a reasonable level through the *MT*, *CR*, and *CO* adjustments. Afterward, coasting control is adopted over the longer track space. Finally, when turning to the *MB*, the speeds of ① and ② are all less than 47 km/h, and ③ starts braking at a speed of 53.17 km/h, but undergoes several transitions between *MB* and *CO* until stopping. This indicates that for these three trajectories, less kinetic energy is consumed by braking, and more kinetic energy is used to overcome resistance. On the other hand, the three trajectories ④, ⑤, and ⑥ have a higher frequency of switching operations. In a relatively long distance of the track, the train maintains high speed through *MT* and *CO*. After switching to continuous *CO*, the distance traveled is shorter. The speed of three trajectories exceeds 54 km/h when switching to *MB*. This means that more energy is consumed by braking. Therefore, the energy consumption of three trajectories ①, ②, and ③ is lower than ④, ⑤, and ⑥, respectively.
3. This is also supported by Table 5, in which three important indicators the energy consumption E_N(kw h), the total trip time T_N(s), and the time deviation $J_{\Delta t} = |T_N - T_P|$(s) are used to evaluate the train operation trajectory. The energy consumption corresponding to the train trajectories obtained by the S-PPO algorithm with the three rewards is 21.13 kw h, 21.47 kw h and 22.06 kw h, and their deviation in trip time are also very small, with values of 0.06 s, 0.07 s and 0.21 s. The PPO algorithm yields energy consumption of 22.32 kw h, 22.41 kw h and 22.61 kw h, with trip time deviations of 0.04 s, 0.57 s and 0.02 s, respectively. The time deviations of the control strategies obtained by S-PPO

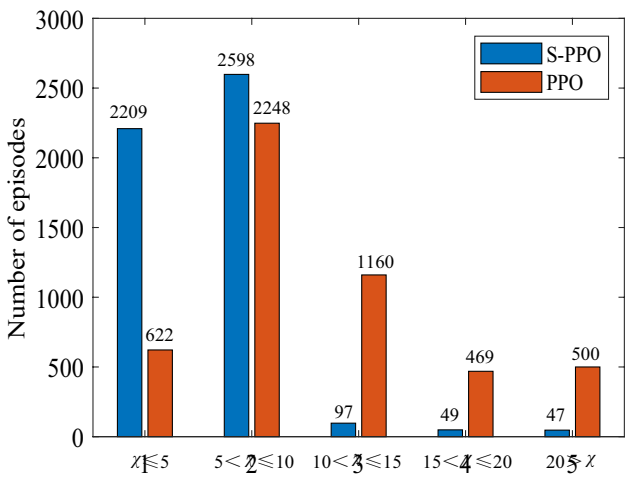

Fig. 11 The result of unsafe action counts of S-PPO and PPO

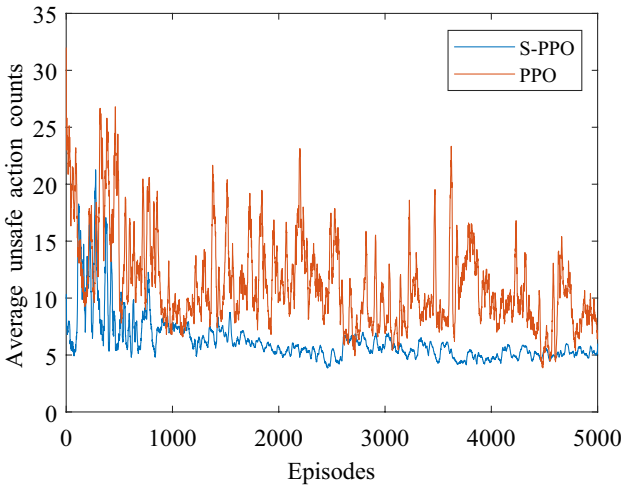

Fig. 12 The comparison of average unsafe action counts of S-PPO and PPO

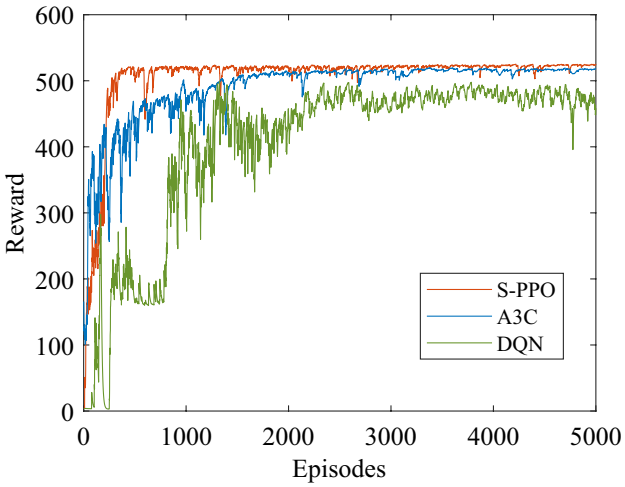

Fig. 13 The convergence curve comparsion among the three algorithm

and PPO with different rewards are less than 1 s, which is within the allowable range. Compared to PPO, S-PPO with relaxed dynamic reward, average dense reward and sparse reward save energy consumption by 5.63%, 4.37%, and 2.49%, respectively. Especially with the combination of S-PPO and relaxed dynamic reward, there is a considerable improvement in energy efficiency on a line with a total length of 1975 m. This further demonstrates that algorithms with relaxed dynamic rewards have better exploration ability than do those with sparse and average dense rewards.

4. The statistical analysis bar chart of the average unsafe action counts of S-PPO and PPO ten experiments is shown in Fig. 11, where parameter χ is the unsafe action counts in each episode. It can be seen that S-PPO has a low level of unsafe action counts ($\chi \leq 5$) in 2209 episodes, accounting for 44.18%, much higher than PPO's 622 episodes (12.44%). When $5 < \chi \leq 10$, S-PPO has slightly higher episodes than PPO, with 2598 episodes (51.96%) and 2248 episodes (44.96%), respectively. At high level unsafe action counts ($10 < \chi \leq 15$, $15 < \chi \leq 20$ and $\chi > 20$), the number of episodes for S-PPO is much lower than PPO. As a result, S-PPO maintains a lower unsafe action counts for a episode throughout the entire iteration process, while PPO does the opposite. This viewpoint is also supported by Fig. 12 that is a moving average unsafe action counts curve of ten experiments for S-PPO and PPO. From Fig. 12, it can be seen that the unsafe action counts of PPO remain oscillating at a high level. However, the unsafe action counts of S-PPO experienced severe oscillations before the 1000 episodes and has been able to maintain a relatively low steady state since then.

   The higher proportion of low-level unsafe action counts and the lower proportion of high-level unsafe action counts indicate that the SARM, which can effectively limit the unsafe actions to a lower level and protect the learning process, is superior to the penalty mechanism based soft constraint method.
5. The convergence curves and train operating trajectories of the S-PPO, A3C, and DQN algorithms are shown in Figs. 13 and 14, respectively. As shown in Fig. 13, in comparison with the A3C and DQN algorithms, the S-PPO algorithm

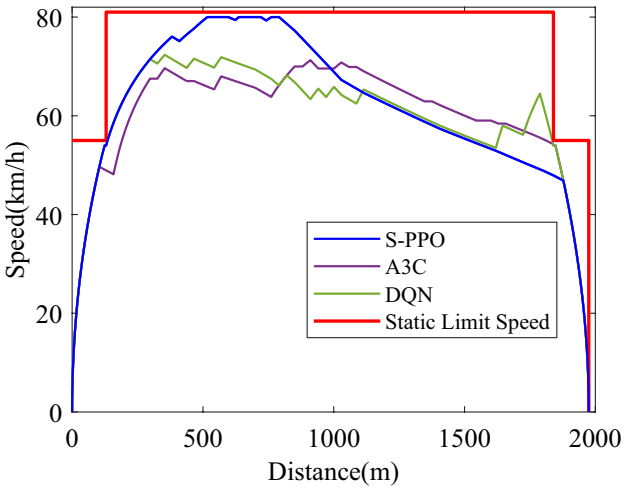

Fig. 14 The trajectories of train operating comparsion among the three algorithm

Table 6 The performance metrics of S-PPO, A3C, and DQN

Performance	S-PPO	A3C	DQN
E_N	21.13	21.65	24.03
T_N	130.06	129.97	130.14
$J_{\Delta t}$	0.06	0.03	0.14

Table 7 Speed limit information for the track

Sequence	Starting point (*m*)	Endpoint (*m*)	Speed limit (*km/h*)
①	543	761	50
②	1115	1405	60
③	851	1029	60
	1029	1115	50
④	700	851	60
	1405	1620	60

exhibits a quicker and smoother convergence, achieving superior reward values. In Fig. 14, the S-PPO algorithm's trajectory minimizes the number of *MT* and *MB*, effectively reducing braking duration through extensive coasting. The performance metrics for S-PPO, A3C, and DQN are presented in Table 6. The S-PPO algorithm achieved a runtime of 130.06 s on the track, maintaining a close tolerance of only 0.06 s from the planned trip time. Its energy consumption is 21.13 kw h, the lowest among the three algorithms. These results suggest that S-PPO exhibits superior performance in TTO problems.

In summary, the combination of the relaxed dynamic rewards and the SARM enhances the exploration and convergence capabilities of the S-PPO algorithm, which helps it obtain better train trajectories in TTO problems.

4.2 Universality test

Given that varying speed limit locations and the shape of speed limit curves significantly affect the trajectories of train operating, we have developed four different speed limit curves to assess the adaptability of S-PPO. The speed limit information is shown in Table 7. ① and ② denote the establishment of a single speed limit at the proximate and terminal positions along the track, respectively. ③ and ④, different speed limit combinations have been adopted. The

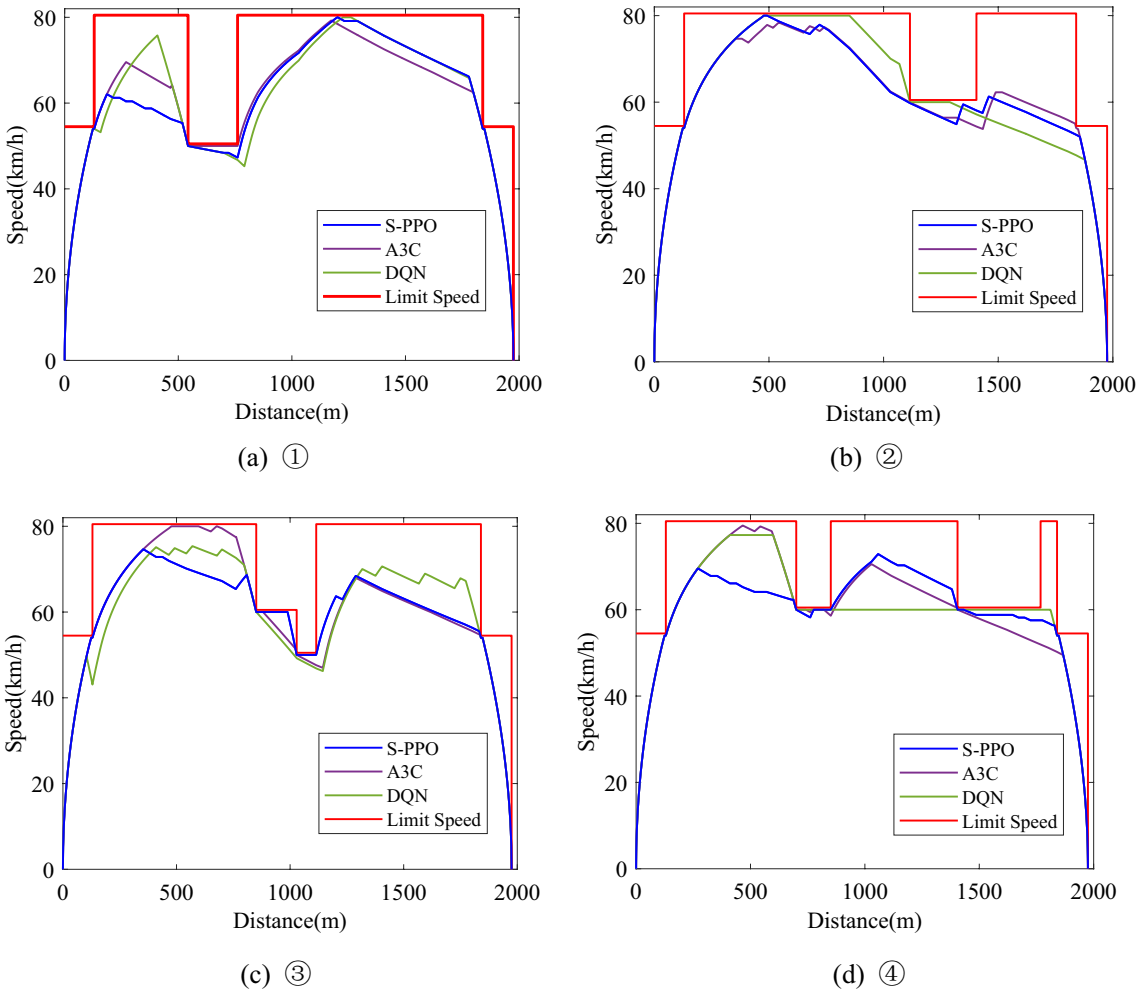

Fig. 15 The trajectories of train operating under various speed limit using S-PPO, A3C, and DQN

Table 8 Performance data under different speed limits

Algorithms	Performance	Speed limits sequence			
		①	②	③	④
S-PPO	E_N	**25.96**	**21.86**	**25.57**	**23.81**
	T_N	130.15	130.07	130.06	130.20
	$J_{\Delta t}$	0.15	**0.07**	0.06	0.20
A3C	E_N	27.37	22.33	26.56	26.01
	T_N	130.13	129.83	129.96	130.16
	$J_{\Delta t}$	**0.13**	0.17	**0.04**	**0.16**
DQN	E_N	31.26	23.30	28.76	27.63
	T_N	130.49	130.07	130.11	129.49
	$J_{\Delta t}$	0.49	**0.07**	0.11	0.51

The best experimental result is shown in bold

trajectories of trains under four different speed limits, achieved using S-PPO, A3C, and DQN algorithms, are depicted in Fig. 15. Correspondingly, detailed performance metrics are compiled in Table 8.

As shown in Fig. 15, train operating trajectories of the S-PPO, A3C, and DQN algorithms on four tracks with varying speed limits are presented. The corresponding performance metrics for these trajectories are outlined in Table 8. The trajectories of S-PPO exhibit consistent characteristics. Prior to reaching or surpassing speed limit starting point,

the train endeavors to increase its speed as much as possible, and then effectively employs *CR* and *CO* to maintain a reasonable speed level, significantly reducing energy losses due to *MB*. This aligns with the features of the train's optimal control sequence. This is consistent with the characteristics of the optimal control sequence of the train. S-PPO operates with energy consumption of 25.96, 21.86, 25.57, and 23.81 respectively on ①, ②, ③, and ④, demonstrating better energy-saving efficiency than A3C and DQN, and has a sufficiently small deviation in operating time ($J_{\Delta t} \leq 0.2s$). The train operation trajectories of A3C on four tracks with varying speed limits has fewer action transitions. Its time deviation on ①, ③, and ④ is superior to that of S-PPO, albeit within a narrow margin of 0.04. Nevertheless, its energy-saving efficiency significantly trails that of S-PPO. DQN has the worst energy-saving effect and operating time deviation.

Therefore, S-PPO can adapt well to different speed limits and obtain satisfactory train operation trajectories, with strong universality.

5 Conclusion

This paper presents an S-PPO algorithm to solve the problem that the soft constraints in reinforcement learning cannot fully protect the learning process from interference and damage caused by actions that exceed safety limits. The SARM has been designed to ensure the agent remains within safe boundaries during the learning process, and combined with the RDRM to enhance the algorithm's exploration ability. The simulation experiment results confirm that these mechanisms significantly improve algorithm stability, making the learning process more efficient while meeting environmental safety constraints. In addition, series of experiments have also demonstrated that the S-PPO performs well under various speed limit conditions and achieves effective train operation strategies, indicating its strong generalization ability and adaptability to energy-saving optimization challenges in diverse environments. It holds broad application prospects in practical scenarios.

In future research, we will continue to explore the adaptability of the S-PPO algorithm under dynamic speed limit conditions. Specifically, we will investigate how to optimize the algorithm's performance to meet speed limit requirements in emergency situations as speed limits continuously change. Additionally, we will also explore the application of the S-PPO algorithm in other areas with security requirements, such as drone navigation and intelligent transportation. Through these studies, we aim to provide more effective and stable methods for the application of reinforcement learning in security-critical areas.

Acknowledgements This work was financially supported by the National Key Research and Development Plan under grant number 2020YFB1713600 and the National Natural Science Foundation of China under grant numbers 62063021. It was also supported by the Key talent project of Gansu Province (ZZ2021G50700016), and Jiangmen Basic and Theoretical Science Research Project, 2023 (2023JC01001), respectively.

Author contributions C.L. wrote the main manuscript text. C.J. has provided significant insights and suggestions that have shaped the direction and focus of the research. Y.X and Z.Z. collected and interpreted the data, ensuring its accuracy and relevance to the research questions. W.W. has been instrumental in the design of the models and algorithms used in our research. Each author provided constructive revision suggestions for the manuscript, and all authors read and approved the submitted manuscript.

Funding This study was funded by Wuyi University

Data availability The track and train basic data used in this study were sourced from Yang Xin's (2016) doctoral thesis "Research on Train Timetable Optimization for Energy-Saving Operations in Urban Rail Transit". We have verified the completeness of the data, and other relevant data are explained in this manuscript to ensure their suitability for the analysis of this study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Guangzhou Metro Group Co., L. Guangzhou metro 2022 annual report. https://www.gzmtr.com/ygwm/gsgk/qynb/202306/P020230728361835232475.pdf.
2. Zhu C, Lu J, Li X. Review of studies on energy-efficient train operation in high-speed railways. IEEJ Trans Electr Electron Eng. 2022;18:451–62. https://doi.org/10.1002/tee.23741.
3. Milroy IP. Aspects of automatic train control. Electronic Thesis or Dissertation, Loughborough University; 1980.
4. Howlett P. Existence of an optimal strategy for the control of a train. School of Mathematices Report 3, Unversity of South Australia; 1988.
5. Howlett P. Optimal strategies for the control of a train. Automatica. 1996;32:519–32. https://doi.org/10.1016/0005-1098(95)00184-0.
6. Liu RR, Golovitcher IM. Energy-efficient operation of rail vechicles. Transp Res Part A Policy Pract. 2003;37:917–32. https://doi.org/10.1016/j.tra.2003.07.001.
7. Howlett P, Pudney P. Energy-efficient train. Control. 1995. https://doi.org/10.1016/0967-0661(94)90198-8.
8. Wang P, Goverde RMP. Multiple-phase train trajectory optimization with signalling and operational constraints. Transp Res Part C Emerg Technol. 2016;69:255–75. https://doi.org/10.1016/j.trc.2016.06.008.
9. Wang P, Goverde RMP. Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. Transp Res Part B Methodol. 2017;105:340–61. https://doi.org/10.1016/j.trb.2017.09.012.
10. He J, Qiao D, Zhang C. On-time and energy-saving train operation strategy based on improved AGA multi-objective optimization. Proc Inst Mech Eng Part F J Rail Rapid Transit. 2023. https://doi.org/10.1177/09544097231203271.
11. Lu G, He D, Zhang J. Energy-saving optimization method of urban rail transit based on improved differential evolution algorithm. Sensors (Basel). 2022. https://doi.org/10.3390/s23010378.
12. Pan Z, Chen M, Lu S, Tian Z, Liu Y. Integrated timetable optimization for minimum total energy consumption of an AC railway system. IEEE Trans Veh Technol. 2020;69:3641–53. https://doi.org/10.1109/tvt.2020.2975603.
13. Cao F, Fan LQ, Tang T, Ke BR. Optimisation of recommended speed profile for train operation based on ant colony algorithm. Int J Simul Process Model. 2016. https://doi.org/10.1504/IJSPM.2016.078512.
14. Ko H, Koseki T, Miyatake M. Application of dynamic programming to the optimization of the running profile of a train. In: Proceedings of the advances in transport; 2004. p. 103–112.
15. Lu S, Hillmansen S, Ho TK, Roberts C. Single-train trajectory optimization. IEEE Trans Intell Transp Syst. 2013;14:743–50. https://doi.org/10.1109/tits.2012.2234118.
16. Liu W, Su S, Tang T, Wang X. A DQN-based intelligent control method for heavy haul trains on long steep downhill section. Transp Res Part C Emerg Technol. 2021. https://doi.org/10.1016/j.trc.2021.103249.
17. Yin J, Chen D, Li L. Intelligent train operation algorithms for subway by expert system and reinforcement learning. IEEE Trans Intell Transp Syst. 2014;15:2561–71. https://doi.org/10.1109/tits.2014.2320757.
18. Kouzoupis D, Pendharkar I, Frey J, Diehl M, Corman F. Direct multiple shooting for computationally efficient train trajectory optimization. Transp Res Part C Emerg Technol. 2023. https://doi.org/10.1016/j.trc.2023.104170.
19. Liu T, Xun J, Yin J, Xiao X. Optimal train control by approximate dynamic programming: comparison of three value function approximation methods. 2018. p. 2741–2746. https://doi.org/10.1109/ITSC.2018.8569440.
20. Wang P, Trivella A, Goverde RMP, Corman F. Train trajectory optimization for improved on-time arrival under parametric uncertainty. Transp Res Part C Emerg Technol. 2020. https://doi.org/10.1016/j.trc.2020.102680.
21. Liu R, Li S, Yang L, Yin J. Energy-efficient subway train scheduling design with time-dependent demand based on an approximate dynamic programming approach. IEEE Trans Syst Man Cybern Syst. 2020;50:2475–90. https://doi.org/10.1109/tsmc.2018.2818263.
22. Watkins CJCH, Dayan P. Q-learning. Mach Learn. 1992;8:279–92. https://doi.org/10.1007/bf00992698.
23. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature. 2015;518:529–33. https://doi.org/10.1038/nature14236.
24. Wu Y, Liao S, Grosse R, Ba J, Mansimov E. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. arXiv; 2017.
25. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning; Comput Sci; 2015. https://doi.org/10.1016/S1098-3015(10)67722-4.
26. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv; 2017. https://doi.org/10.48550/arXiv.1707.06347.
27. Liang H, Zhang Y. Research on automatic train operation performance optimization of high speed railway based on asynchronous advantage actor-critic. In: Proceedings of the 2020 Chinese Automation Congress (CAC); 2020. p. 1674–80.
28. Zhang L, Zhou M, Li Z. An intelligent train operation method based on event-driven deep reinforcement learning. IEEE Trans Ind Inf. 2022;18:6973–80. https://doi.org/10.1109/tii.2021.3138098.
29. 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. 2022;52:716–27. https://doi.org/10.1109/tsmc.2020.3000073.
30. Pang Z, Wang L, Li L. A hybrid machine learning approach for train trajectory reconstruction under interruptions considering passenger demand. Int J Rail Transp. 2024. https://doi.org/10.1080/23248378.2024.2329717.
31. García J, Fernández F. A comprehensive survey on safe reinforcement learning. J Mach Learn Res. 2015;16:1437–80.
32. Sui Y, Gotovos A, Burdick JW, Krause A. Safe exploration for optimization with Gaussian processes. JMLR.org. 2015.
33. Turchetta M, Berkenkamp F, Krause A. Safe exploration in finite Markov decision processes with Gaussian processes; 2016. https://doi.org/10.48550/arXiv.1606.04753.

34. Wachi A, Kajino H, Munawar A. Safe exploration in Markov decision processes with time-variant safety using spatio-temporal Gaussian process; 2018. https://doi.org/10.48550/arXiv.1809.04232.
35. Alshiekh M, Bloem R, Ehlers R, Könighofer B, Niekum S, Topcu U. Safe reinforcement learning via shielding; 2017. https://doi.org/10.48550/arXiv.1708.08611.
36. Jeddi AB, Dehghani NL, Shafieezadeh A. Memory-augmented Lyapunov-based safe reinforcement learning: end-to-end safety under uncertainty. IEEE Trans Artif Intell. 2023;4:1767–76. https://doi.org/10.1109/TAI.2023.3238700.
37. Zhou Z, Oguz OS, Leibold M, Buss M. Learning a low-dimensional representation of a safe region for safe reinforcement learning on dynamical systems. IEEE Trans Neural Netw Learn Syst. 2023;34:2513–27. https://doi.org/10.1109/TNNLS.2021.3106818.
38. Mataric MJ. Reward functions for accelerated learning. In: Machine learning proceedings; 1994. p. 181–9.
39. Luo Y, Wang Y, Dong K, Liu Y, Sun Z, Zhang Q, Song B. D2SR: transferring dense reward function to sparse by network resetting. In Proceedings of the 2023 IEEE international conference on real-time computing and robotics (RCAR); 2023. p. 906–11.
40. Andrychowicz M, Wolski F, Ray A, Schneider J, Fong R, Welinder P, Mcgrew B, Tobin J, Abbeel P, Zaremba W. Hindsight Experience Replay. arXiv; 2017.
41. Plappert M, Andrychowicz M, Ray A, Mcgrew B, Baker B, Powell G, Schneider J, Tobin J, Chociej M, Welinder P. Multi-goal reinforcement learning: challenging robotics environments and request for research; 2018. https://doi.org/10.48550/arXiv.1802.09464.
42. Manela B, Biess A. Curriculum learning with hindsight experience replay for sequential object manipulation tasks. Neural networks: the official journal of the International Neural Network Society. 2022;145:260-270. https://doi.org/10.1016/j.neunet.2021.10.011.
43. Vecerik M, Hester T, Scholz J, Wang F, Pietquin O, Piot B, Heess N, Rothrl T, Lampe T, Riedmiller M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards; 2017. https://doi.org/10.48550/arXiv.1707.08817.
44. Ng AY, Harada D, Russell S. Policy invariance under reward transformations: Theory and application to reward shaping. Morgan Kaufmann Publishers Inc.; 1999.
45. Wang J, Liu Y, Li B. Reinforcement learning with perturbed rewards; 2018. https://doi.org/10.1609/aaai.v34i04.6086.
46. He Q, Hou X. WD3: taming the estimation bias in deep reinforcement learning. In: Proceedings of the 2020 IEEE 32nd international conference on tools with artificial intelligence (ICTAI); 2020.
47. Albrecht A, Howlett P, Pudney P, Vu X, Zhou P. The key principles of optimal train control—Part 1: formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. Transp Res Part B. 2016;94:482–508.
48. Zhao Z, Xun J, Wen X, Chen J. Safe reinforcement learning for single train trajectory optimization via shield SARSA. IEEE Trans Intell Transp Syst. 2023;24:412–28. https://doi.org/10.1109/tits.2022.3218705.
49. Ma S, Ma F, Tang C. An energy-efficient optimal operation control strategy for high-speed trains via a symmetric alternating direction method of multipliers. Axioms. 2023. https://doi.org/10.3390/axioms12050489.
50. Zhu Q, Su S, Tang T, Liu W, Zhang Z, Tian Q. An eco-driving algorithm for trains through distributing energy: A Q-Learning approach. ISA Trans. 2022;122:24–37. https://doi.org/10.1016/j.isatra.2021.04.036.
51. Ye H, Liu R. Nonlinear programming methods based on closed-form expressions for optimal train control. Transp Res Part C Emerg Technol. 2017;82:102–23. https://doi.org/10.1016/j.trc.2017.06.011.
52. Lin X, Liang Z, Shen L, Zhao F, Liu X, Sun P, Cao T. Reinforcement learning method for the multi-objective speed trajectory optimization of a freight train. Control Eng Pract. 2023. https://doi.org/10.1016/j.conengprac.2023.105605.
53. Huang J, Zhang E, Zhang J, Huang S, Zhong Z. Deep reinforcement learning based train driving optimization. :In Proceedings of the 2019 Chinese Automation Congress (CAC); 2019.